



Operator's Manual
7070 Series Programming Systems:
Compiler Systems

IBM

IBM

Operator's Manual
7070 Series Programming Systems:
Compiler Systems

This manual is a revision of, and a replacement for, the following publications:

IBM 7070/7074 Data Processing System Bulletin "IBM 7070/7074
Compiler Systems: Operating Procedure," form J28-6105

IBM 7070/7074 Data Processing System Bulletin "IBM 7070/7074
Compiler Systems: Operating Procedure - Supplement," form J28-6185-0

IBM 7070 Series Data Processing Systems Bulletin "IBM 7070 Series
Programming Systems: COBOL Processor (7070/7074) - Operating
Instructions," form J28-6189-0

IBM 7070 Series Data Processing Systems Bulletin "IBM 7070 Series
Programming Systems: Commercial Translator Processor (7070/7074) -
Operating Instructions," form J28-6179

IBM Technical Newsletter "Composition of the 7070 Series Compiler
Systems Tape," form N28-0021.

It also contains the diagnostic messages formerly included in the IBM 7070
Series Data Processing Systems Bulletin "IBM 7070 Series Programming Systems:
COBOL Processor (7070/7074)," form J28-6182.

CONTENTS

Introduction	1
General Fundamentals	2
Run Types	2
Operating Options	2
Control Cards	2
Input/Output Formats	3
Input/Output Unit Allocations	4
Tapes	4
Options for Input	4
Input/Output Unit Requirements	4
Options for Output	4
Operating Instructions	6
Compile Run	6
Systems Run	8
Generator Run	12
Multifile Run	15
Control Cards: Listing and Description	19
RUN Control Cards	19
Operating Option Control Cards	19
Processor Machine	21
Object Machine	23
Permanent Operating Option Changes	26
Systems Update Control Cards	28
Operation and Installation Suggestions	29
Programmed Halts and Messages	30
Messages	38
Appendix A: Checkpoint in COBOL and Commercial Translator	42
Appendix B: Loading the FORTRAN Object Program	43
Appendix C: Report Program Operating Procedures	50
Appendix D: Composition of the Compiler Systems Tape	51
Appendix E: Communications Record	63
Appendix F: Glossary	65

INTRODUCTION

The compilers on the 7070 Series Compiler Systems Tape constitute a complete package which allows a user to move from problem definition to productive computer usage and problem solution with a minimum of programming effort. These compilers are designed as interlocking components, with the other processors making use of major sections of Autocoder in translating their respective input data into machine-language instructions.

The systems tape provides for the automatic translation of the powerful Autocoder programming language into machine-language instructions to be executed by the IBM 7070, 7072, or 7074. It also provides for the compilation of programs written in COBOL, Commercial Translator, or FORTRAN language, and for the generation of complete programs from the specifications furnished to the Report Program Generator.

This manual is designed for the person who operates the IBM 7070, 7072, or 7074 during the use of the Compiler Systems Tape. It will also be of assistance to those who prepare input data for later processing by the system. While knowledge of the languages and specifications used by Autocoder, COBOL, Commercial Translator, FORTRAN, and the Report Program Generator is desirable, it is not a prerequisite to the use of this manual. Further information regarding these processors can be found in the following publications:

1. IBM General Information Manual "COBOL", form F28-8053-1.
2. IBM 7070 Series Data Processing Systems Bulletin "IBM 7070 Series Programming Systems: COBOL Processor (7070/7074)", form J28-6182-1.
3. IBM General Information Manual "Commercial Translator", form F28-8043.
4. Addenda to the IBM Commercial Translator General Information Manual, form J28-8072.

5. "IBM 7070 Series Programming Systems: Commercial Translator Processor (7070/7074)", Preliminary Bulletin, form J28-6180.

6. IBM Reference Manual "IBM 7070 Series Programming Systems: Autocoder," form C28-6121-0.

7. IBM Reference Manual "IBM 7070/7074 Compiler Systems: Report Program Generator," form C28-6113

8. IBM Reference Manual "IBM 7070 Series Programming Systems: FORTRAN," form C28-6170.

9. IBM General Information Manual "FORTRAN," form F28-8074.

It is assumed in this publication that the user is familiar with the IBM Reference Manual "7070 Data Processing System," form A24-7003-2.

The early sections of this manual discuss the general philosophy of the systems tape and the various machine runs for which the tape may be used. Later sections include specific operating procedures for each run, loading instructions for FORTRAN object programs and Report Programs, and other information.

The Compiler Systems Tape may be obtained by sending a full reel of magnetic tape to the Manager, DP Library Services. The compilers will be written on the tape, which will then be returned. It is suggested that a duplicate of the tape be made (by a Systems Run) as soon as it is received so that one copy of the program can be kept in reserve. The extra copy should be used only if the working copy becomes unusable. The reel of magnetic tape for the 7070 Series Compiler Systems Tape should be sent to:

DP Library Services
IBM Programming Systems
112 East Post Road
White Plains, N. Y.

GENERAL FUNDAMENTALS

RUN TYPES

The Compiler Systems Tape may be used for a variety of purposes, referred to as "runs." These fall into three general categories: Compile Runs, Systems Runs, and Generator Runs.

Compile Runs

During compilation, source-language statements (written in Autocoder, COBOL, Commercial Translator, or FORTRAN language, or as Report Program Generator specifications) are converted into machine-language instructions. In addition, a symbolic-language listing of the input data and associated machine instructions is produced. The original compilation of a source-language program is accomplished by the most common run, a Compile Run. If assembly of two or more programs is required, a Multifile Run may be desirable.

Systems Runs

Since extensibility is one of the important attributes of the languages compiled by the Compiler Systems Tape, provision is made for runs which create a new systems tape, into which are incorporated any necessary changes. During a Systems Run, changes are processed against the current systems tape to produce a separate, updated systems tape. A record of the changes is also produced as an aid to the person who maintains control over the systems tape contents.

Generator Runs

In the Autocoder language represented on the Compiler Systems Tape, provision is made for processing "macro-instructions." These are single statements by the programmer which cause the generation of multiple machine-language instructions. A Generator Run compiles input statements which create a new macro generator, i. e., a section of the systems tape that will process a specific macro-instruction. Since this run produces a program listing for the compiled macro generator and, if desired, enters the resultant program on a new systems tape, it can perform the combined functions of a Compile Run and a Systems Run.

OPERATING OPTIONS

The individual components within an IBM 7070, 7072, or 7074 Data Processing System may vary from installation to installation. For example, one system

may be tape-oriented, while another may use an on-line printer; or one system may have six tape units, while another has ten tape units. In addition, the machines may differ as to size of core storage, ability to do floating-decimal arithmetic, etc. In this manual, these variables or machine "characteristics" are referred to as "operating options" and are considered as they apply to three general areas:

1. The "processor" machine on which the systems tape is to be operated.
2. The "object" machine on which the resultant systems tape or compiled program is to be operated.
3. Various considerations concerning input and output, such as presence or absence of tape labels, density, etc.

In most installations, the object and processor machines will be the same; however, the Compiler Systems Tape is designed to allow consideration of them as separate machines with separate sets of characteristics. Thus, a program may be compiled on a one-channel system for later use on a system with two channels; a program may be compiled on a machine with 5,000 words of core storage for use on a machine with 9,990 words of core storage, etc.

The systems tape is supplied with certain "permanent," or built-in, operating options. However, these options are modifiable in whole or in part, according to the current needs of a particular installation. Such modifications may be temporary, i. e., for a specific run, or they may be made a permanent part of a new systems tape.

CONTROL CARDS

Control cards are used to designate language, machine characteristics, and the type of run to be performed. These cards are discussed in more detail in later sections of this manual. They are referred to as follows:

1. RUN Control Cards which specify the current run type (e. g., COMPILE RUN, SYSTEMS RUN, etc.) and may also identify the input source language.
2. Operating Option Control Cards which perform two functions:
 - a. When placed before any RUN Control Card, they affect the operating options for the current run only.
 - b. When placed after a SYSTEMS RUN Control Card, they cause the permanent operating options to be redefined during the creation of a new systems tape.
3. Systems Update Control Cards which are used during a Systems Run to indicate other changes

to be considered in the creation of a new systems tape. Such changes include the addition, deletion, or modification of information on the systems tape.

The use of the above control cards allows unusual flexibility in the utilization of the Compiler Systems Tape. However, once permanent operating options are incorporated into the systems tape so that the most effective use is made of equipment at an installation, the RUN Control Card is the only one required for most runs.

INPUT/OUTPUT FORMATS

Input for a typical run consists of one or more control cards followed by the cards containing the input data. Control cards and source program input data written in Autocoder language should be punched into IBM 7070 Autocoder Input Cards (IBM electro A18265). For programs written as Report Program Generator specifications, a general purpose card (IBM electro 5081) is recommended. Source programs written in FORTRAN language should be punched in FORTRAN Statement cards (IBM electro 888157). Source programs written in COBOL language should be punched in COBOL Source Program Cards (IBM electro C61897). Source programs written in Commercial Translator language should be punched in Commercial Translator cards (IBM electro C58810 and C58811).

Punched card input may be read directly using an on-line card reader, or it may be written on tape by an off-line card-to-tape operation and then read from the tape. A combination of card and tape input may be used for a given run.

Several types of output may be obtained from a run using the Compiler Systems Tape. A new systems tape (or a copy of the present one) is always produced by a Systems Run. An object program deck and associated symbolic language listing are always produced by a Compile Run. A compiled Generator program deck and a symbolic listing of the compiled macro generator are always produced by a Generator Run, and, if desired, a new systems tape (including the macro generator just compiled) is created.

When specified, the first five cards produced during a run contain the IBM 7070/7074 Condensed Card Load Program. When punched, the object program is in condensed card form, i. e., it has a maximum of five machine-language instructions or constants per card. The object program should be punched into IBM 7070 Condensed Cards (IBM electro C54549). The program listing produced by the compiler includes the data from the source input card, the resulting machine-language instruction, the assigned location, any messages generated by the compiler to indicate suspected errors, etc.

INPUT/OUTPUT UNIT ALLOCATION

TAPES

Runs made using the Compiler Systems Tape may utilize from two through ten tape units. The Compile Run, which is used most frequently, requires a minimum of six tape units for Autocoder, FORTRAN, and Report Program Generator programs, and a minimum of seven tape units for COBOL and Commercial Translator programs. The distribution of these units is indicated in Table I. The symbolic names are provided for purposes of simplifying later references to the various tape units.

TABLE I

Tape Unit	1 Channel Operation Tape Channel & Unit	2 Channel Operation Tape Channel & Unit
SYSTEMTAPE	10	10
WORKTAPEA	11	11
WORKTAPEB	12	12
WORKTAPEC	13	20
WORKTAPE D	14	21
WORKTAPEE	15	22
WORKTAPEF (1)	(2)	25

- (1) For COBOL and Commercial Translator.
 (2) Any tape unit on channel 1 other than units 0 through 5.

As many as four additional tape units may be required if the user elects certain options concerning input and output which are discussed below. Operating option control cards are used to name the added tape units which are available.

OPTIONS FOR INPUT

For added flexibility, input consists of two groups of cards.

The first group, which will be referred to as "initial" input, contains:

1. The RUN Control Card.
2. Any operating option control cards which may precede it.

The second group, referred to as "main" input, usually consists of the source program data cards. In a Systems Run, however, the main input includes any control cards which may follow the RUN Control Card (e.g., operating option, systems update), together with associated condensed load cards; in a Multifile Run, the COMPILE RUN Control Card for each file is part of the main input.

Initial input may be read from either of two input units, referred to by the symbolic names INITAUNIT and INITBUNIT. During a run, the input unit to be used is selected by means of console alteration switch 1. If this switch is OFF, initial input will be read from INITAUNIT; if it is ON, initial input will be read from INITBUNIT. Either a card reader or a tape unit may be designated as either input unit at the time the

Compiler Systems Tape is created. If a card reader is designated as one input unit and a tape unit is designated as the other, the initial input for any run may be either punched cards or tape.

MAINUNIT is the symbolic name of the card reader or tape unit from which main input will be read. If MAINUNIT is a tape unit, it may be the same unit as WORKTAPEA, but it may not be any other work tape named in Table I. If more than six tape units are available, MAINUNIT may be one of these additional units.

Initial and main input may be in the form of punched cards and/or card image tapes. The choice as to which form to use may be limited by the availability of input equipment on the processor machine. The input/output requirements for all runs using the Compiler Systems Tape are outlined in Table II.

Further references in this manual to the symbolic names used above will be followed by parentheses containing the number of the input device designated on the systems tape supplied by IBM. It should be understood, however, that the user is free to alter the designation of input units so that the most effective use is made of the equipment at a particular installation.

INPUT/OUTPUT UNIT REQUIREMENTS

The input and output requirements for each type of run are outlined in Table II.

TABLE II

Type of Run	729 Model II or IV Tape Units.			IBM 7500 or 7501 Card Reader (See Note as Indicated)	IBM 7550 Card Punch for on-line punching of object deck	IBM 7400 Printer for on-line printing of program listing
	Minimum	Additional (See Note as indi- cated)	Maximum			
COMPILE	6	A	8	C	Optional	Optional
MULTIFILE	7	B	10	C	See Note D	Optional
SYSTEMS	2	A	4	C	No	No
GENERATOR	6	A	8	C	Optional	Optional

NOTES:

- A. One additional tape unit is required if either of the following is true:
 1. There is input data on a tape other than WORKTAPEA (tape 11).
 2. The source program language is COBOL or Commercial Translator.
- Two additional tape units are required if initial input and main input are on separate tapes, neither of which is WORKTAPEA (tape 11).
- B. For a Multifile Run only, one additional tape unit is required if any input is on tape. A second additional tape unit is required if initial input data and main input data are on separate tapes.
- Another additional tape unit, DECKTAPE, is required if the object program deck is to be written on tape, and the MIXEDTAPE option is not elected.
- C. The card reader is required if there is any punched card input.
- D. For a Multifile Run only, the object program deck must be punched on-line if the MIXEDTAPE option is not elected, and the additional tape required for writing the object program deck is not available.

OPTIONS FOR OUTPUT

The output that may be obtained from a run depends on the availability of output equipment. If

the input and output unit requirements outlined in Table II are met, the output that may be obtained for each source program is shown in Table III.

TABLE III

Type of Run	Object Program Listing Tape	Condensed Card Object Program Tape	On-Line Object Program Listing	On-Line Object Program Deck	New Systems Tape
COMPILE MULTIFILE	Always Always	Always Optional	Optional Optional	Optional Mandatory if no object tape; otherwise, optional	No No
SYSTEMS GENERATOR	No Always	No Always	No Optional	No Optional	Always Optional

The console typewriter is used for all halts and messages that are of concern to the operator.

COBOL or Commercial Translator diagnostic messages are written on an error tape and can, at

the user's option, be printed on-line after compilation has occurred.

If alteration switch 3 is ON during a run, additional messages of an informative nature will be typed. These include:

1. The version and change level of the systems tape.
2. Sort output totals.
3. Tape End-of-File messages.
4. If a new systems tape has been created during a Systems Run or a Generator Run, the full ten-character name of each coding block written on the new tape.

If alteration switch 4 is ON during a run, the last five characters of the name of each coding block will be typed as it is read into storage.

During a Systems Run, the typewriter is also used to list all changes which are being made during the production of the new systems tape.

OPERATING INSTRUCTIONS

The instructions for each run are accompanied by an illustration which indicates all possible configurations for the input data and final output of the run. The specific instructions devoted to each run are presented according to the following assumptions concerning the user's installation:

1. The characteristics of the processor and object machines are the same as those specified on the systems tape as supplied by IBM.
2. Initial input can be either punched cards or tape.
3. Main input will be a card-image tape.
4. If both initial input data and main input data are on tape, they are on the same tape.

When an input tape is prepared off-line, all cards must be written on tape in the form of sixteen-word records; all words must be alphameric.

When on-line unit record equipment is used with any type of run, an IBM utility panel must be used in the unit record machine. The wiring of IBM utility panels is described in the 7070 Data Processing System Bulletin "IBM 7070 Utility Control Panels," form J28-6095.

COMPILE RUN

Description

The Compile Run processes symbolic source-language input data to produce a machine-language program deck and a program listing.

The minimum input for this run consists of:

1. The COMPILE RUN Control Card.
2. The cards containing the source-language input data to be compiled.

One of the following COMPILE RUN Control Cards must be used for every Compile Run:

Line	Label	Operation	OPERAND														
			3	5/6	15/16	20/21	25	30	35	40	45						
01	COMPILE	RUN	A	U	T	O	C	O	D	E	R						
02	COMPILE	RUN	C	O	B	O	L										
03	COMPILE	RUN	C	O	M	M	E	R	C	I	A	L	T	R	A	N	
04	COMPILE	RUN	F	O	R	T	R	A	N								
05	COMPILE	RUN	R	E	P	O	R	T	G	E	N	E	R	A	T	O	R
06																	

Lines 01 through 04 show the control cards used for input data in the languages indicated in the operand column. Line 05 shows the control card used for input data in the form of Report Program Generator specifications.

The type of run is determined by the contents of columns 21, 22, and 23 only; however, the additional characters will appear in the typewriter listing.

Input for a Report Program Generator run must be arranged in the following sequence:

1. The Type Code 1 card.
2. Any number of Type Code 2 cards.
3. Any number of Type Code 3 cards.
4. Any number of Type Code 4 cards.
5. An End card, i.e., a card having the word END punched in columns 6 through 8 and all other columns blank.

If a COBOL or Commercial Translator compilation is to be performed and the Environment Division of the source program indicates that checkpoints are to be taken, see "Checkpoint: COBOL and Commercial Translator," Appendix A.

Operating Procedure for a Compile Run

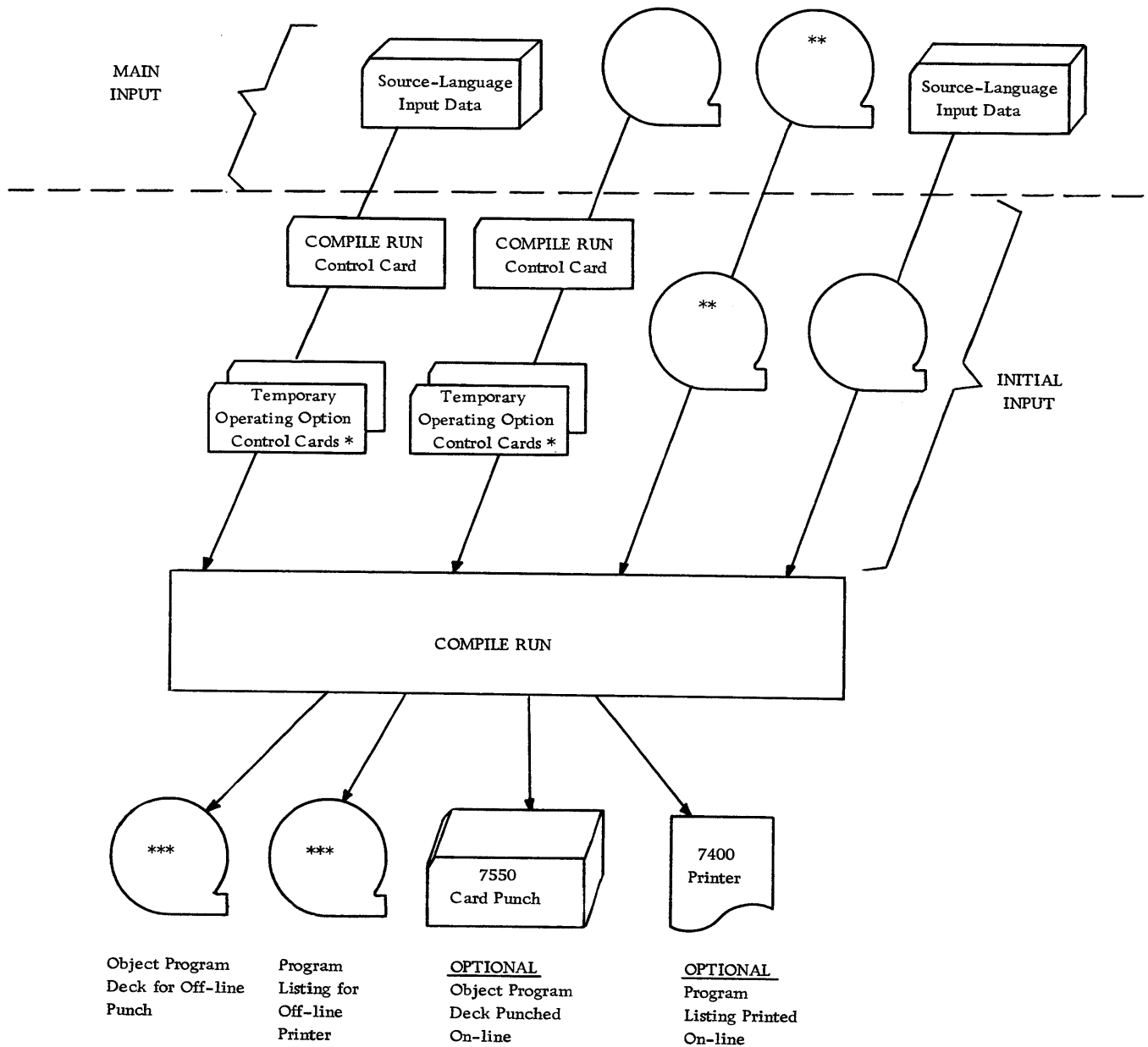
Ready the indicated input/output units as follows:

1. Mount the Compiler Systems Tape on SYSTEMTAPE (tape unit 10). Set the tape unit to high density.
2. If the initial input data is in punched cards:
 - a. Set card reader alteration switches to BBAA.
 - b. Insert the IBM 7500 Utility Panel.
 - c. Load initial input control cards in INITAUNIT (card reader).
 - d. Mount the main input data tape on MAINUNIT (tape unit 11).
3. If the initial input data is on tape: Mount the tape containing initial and main input on INITBUNIT (tape unit 11).
4. Mount work tapes on the tape units specified below for a one-channel or a two-channel operation:

<u>Tape Unit</u>	<u>One Channel</u>	<u>Two Channels</u>
WORKTAPEB	tape unit 12	tape unit 12
WORKTAPEC	tape unit 13	tape unit 20
WORKTAPED	tape unit 14	tape unit 21
WORKTAPEE	tape unit 15	tape unit 22
WORKTAPEF (1)	(2)	tape unit 25

- (1) For COBOL and Commercial Translator compilations.
- (2) Any tape unit on channel 1 other than tape units 1 through 5.

Optional Configurations for Input Data and Final Output



* This input is necessary only if the characteristics of the processor or object machines are to be altered for the current run.

** Main input and initial input may be on the same tape or on separate tapes mounted on separate units.

*** Program listing and object program deck may be on the same tape or on separate tapes mounted on separate units.

5. If the program listing is to be printed on-line (not applicable to COBOL or Commercial Translator):

- a. Set printer alteration switches to AAAA.
- b. Insert the IBM 7400 Utility Panel.

6. If the object program deck is to be punched on-line:

- a. Set punch alteration switches to:
AAAA for a 12-punch in column 65.
ABAA for a 12-punch in column 79.
BAAA for no 12-punch in either column.
- b. Insert the IBM 7550 Utility Panel.
- c. Load punch hopper with cards (IBM electro C54549).

7. If the source program is COBOL or Commercial Translator, and diagnostic messages are to be printed on-line (after compilation), set the printer alteration switches to BBAA.

Store the following instructions manually:

<u>Location</u>	<u>Instruction</u>
0000	-8101010004
0001	-0100030010
0002	+5100110002

Set console controls for operation as follows:

1. Set alteration switches as follows:

Switch 1

Set OFF to read initial input from INITAUNIT (card reader).
Set ON to read initial input from INITBUNIT (tape unit 11).

Switch 2

Set OFF.

Switches 3 and 4

These alteration switches may be used to cause the typing of optional messages (see page 5).

2. Set accumulator overflow keys ON.
3. Set Priority Channel A and Priority Channel B dials.

- a. Set right-hand dials to OFF.
- b. Set left-hand dials to N.

4. Press RUN.
5. Press COMPUTER RESET.
6. Press START.

Anticipated Halts:

Halt 0003 This halt indicates that the main input tape on MAINUNIT (tape unit 11) has been read. This tape may be replaced by a work tape. (See "Programmed Halts and Messages," page 30.)

Halt 0076 This halt indicates completion of the run. A message is typed identifying the tape units which contain the output for the run. In COBOL and Commercial Translator compilations, this halt occurs following the generation of the output tape. Press START to continue through an Autocoder compilation, or save the output tape to be used as input to a separate Autocoder compilation. If the latter procedure is followed, use the INLABELS control card that indicates that the input tape labels are written in low density in connection with the Autocoder compilation. Note, too, that if the Autocoder input tape is to be used as a work tape, the labels, if any, of all other work tapes must be in low density.

Note: The COBOL and Commercial Translator processors use location 0000, which is where the address of the load program is normally stored.

SYSTEMS RUN

Description

The output of a Systems Run is always a new systems tape. This new tape may be created under the following circumstances:

A. A copy of the original systems tape is desired. The Systems Run in this case will merely be a duplicating process. The only input required is the SYSTEMS RUN Control Card.

B. It is necessary to change the permanent operating options from those that exist on the current systems tape. This would normally be true when the original systems tape is received from IBM or when changes are made to the machine configuration at a particular installation.

Input for this run consists of:

- 1. Temporary operating option control cards (if needed).
- 2. The SYSTEMS RUN Control Card.
- 3. Permanent operating option control cards which specify the new operating characteristics to be included in the updated systems tape.

C. It is necessary to incorporate a new coding block in the systems tape or to add to, revise, or delete any of the instructions or data contained in the existing coding blocks. Normally, systems tape modifications will originate from IBM, based on continued testing and analysis of customer experience. However, since macro generators

can be added to the systems tape by the user, revisions and improvements that may be necessary can also be made by means of a Systems Run.

The input for this run consists of:

1. Temporary operating option control cards (if needed).
2. The SYSTEMS RUN Control Card.
3. Systems update control cards (accompanied by condensed load cards when necessary) in ascending collating sequence by the name of the coding block affected. See below.

D. A Systems Run may combine the features of B and C above, i. e., a single run may be used both to change the operating options and to make revisions in the records which constitute the systems tape.

The input for this run consists of:

1. Temporary operating option control cards (if needed).
2. The SYSTEMS RUN Control Card.
3. Permanent operating option control cards.
4. Systems update control cards (accompanied by condensed load cards when necessary) in ascending collating sequence by the name of the coding block affected. See below.

If a duplicate of the original systems tape is desired, use the following RUN Control Card:

Line	Label	Operation	OPERAND							Base
5	36	15	20	25	30	35	40	45	50	
01	S Y S T E M S	R U N	C O P Y							
02										

If changes are to be made which will cause an updated systems tape to be produced, use the following RUN Control Card:

Line	Label	Operation	OPERAND							Base
5	36	15	20	25	30	35	40	45	50	
01	S Y S T E M S	R U N								
02										

Official IBM patches and additions are entered on a user's Compiler Systems Tape under a protective scheme to ensure that the tape being updated is, in fact, a tape to which the patch or addition applies.

The RUN Control Card that accompanies the control cards and condensed load cards needed to update the systems tape has the following format:

Line	Label	Operation	OPERAND							Base
5	36	15	20	25	30	35	40	45	50	
01	S Y S T E M S	R U N	I B M	X	X	X	X	X	X	
02										

In this Run Control Card:

1. IBM identifies this run as an official run.

2. XXXXXX is the version name which is compared against the name stored on the user's systems tape. The run is refused if the names do not match.

3. YYYYYY, if non-blank, replaces the old name XXXXXX on the user's systems tape. If non-blank, it also affects the change level discussed in item 4 below. The version identification changes only on some major revision of the contents of the systems tape.

4. DDDDD is the change level number which is compared against that on the user's systems tape. If the numbers do not match, a halt occurs, but the run may be resumed by pressing START. The change level number on the user's tape is increased by 1 if columns 31 through 35 are blank, or is set to 00001 if these columns are non-blank.

A list of the full ten-character names of all the coding blocks contained on the new systems tape may be obtained during any Systems Run by setting alteration switch 3 ON at the start of the run. This list will be in addition to the typewritten log of changes, if any, made during the creation of the new systems tape.

Systems Update Control Cards

As noted above, systems update control cards are used when it is necessary to change the composition of the coding blocks which constitute the systems tape. These control cards make it possible to drop, patch, or enter coding blocks as necessary.

Only one operation (DROP, PATCH, ENTER) may be performed on a given coding block during any one Systems Run. The Systems Update Control Cards used in a run must appear in ascending collating sequence as determined by the name of the coding block entered in columns 6 through 15.

When a coding block on the old systems tape is not to be included on the new systems tape, use a DROP Control Card punched as follows:

Line	Label	Operation	OPERAND							Base
5	36	15	20	25	30	35	40	45	50	
01	X X X X X X X X	D R O P								
02										

More than one macro generator may be dropped during a non-IBM Systems Run. However, the names of the second through the last macro generators will not be dropped from the macro table. The name of the first macro generator will only be dropped from the table if the DROP Control Card is the first Systems Update Control Card in the input deck for the run. If the names of all the macro generators being dropped are to be removed from the macro table, either the table must be patched or a separate run must be made for each macro generator being dropped.

The name of each macro generator being dropped will be typed during the Systems Run. After the second name is typed, the following message will be issued:

THIS AND ALL FURTHER MACRO CHANGES NOT REFLECTED IN TABLE

For each existing coding block that is to be modified as it is written on the new systems tape, use a PATCH Control Card punched as follows:

Line	Label	Operation	OPERAND					
3	36	15	20	25	30	35	40	45
0.1	XXXXXXXXXXXX	PATCH						
0.2								

Condensed load cards containing additions or revisions must immediately follow the corresponding PATCH Control Card.

For each new coding block that is to be created on the new systems tape, use an ENTER Control Card punched as follows:

Line	Label	Operation	OPERAND					Box	
3	36	15	20	25	30	35	40	45	50
0.1	XXXXXXXXXXXX	ENTER							
0.2									

Condensed load cards to be used in the construction of the new coding block must immediately follow the corresponding ENTER Control Card. If a coding block with the same name already appears on the old systems tape, the new coding block will replace it on the new systems tape.

The IBM 7070/7074 Condensed Card Load Program must not follow the ENTER Control Card. Therefore, if the condensed load cards containing a coding block to be placed on the Compiler Systems Tape are preceded by a load program, the five cards containing the load program must be removed.

More than one compiled macro generator may be entered during a non-IBM Systems Run. However, the names of these macro generators will not be entered in the macro table. Names of macro generators may be entered in the table either by means of a Generator Run, which can only enter one at a time, or by patching the macro table.

An execute card, with a Branch instruction in the first word, may follow the last condensed card for a given coding block. If the current operation is ENTER, the Branch instruction will be used to determine the entrance to the new coding block. If the current operation is PATCH, the Branch instruction will replace the previously determined entrance. If no final execute card appears, the previously determined entrance will be left unchanged.

When an input tape is prepared off-line, all cards (including interspersed condensed load cards) are

to be written on tape in the form of sixteen-word records; all words must be alphameric.

Operating Procedure for a Systems Run

Ready the indicated input/output units as follows:

1. Mount the Compiler Systems Tape on SYSTEMTAPE (tape unit 10).
Set the tape unit to high density.

2. If the initial input data is in punched cards:

a. Set card reader alteration switches to ABAA (for IBM Systems Runs or if a 12-punch in column 79 is to be used for detecting load cards) or to AAAA (if a 12-punch in column 65 is to be used for detecting load cards).

b. Insert the IBM 7500 Utility Panel.

c. Load initial input control cards in INITAUNIT (card reader).

d. Mount the main input data tape on MAINUNIT (tape unit 11).

3. If the initial input data is on tape:

Mount the tape containing initial and main input on INITBUNIT (tape unit 11).

4. Mount a work tape on WORKTAPEB (tape unit 12).

Store the following instructions manually:

<u>Location</u>	<u>Instruction</u>
0000	-8101010004
0001	-0100030010
0002	+5100110002

Set console controls for operation as follows:

1. Set alteration switches as follows:

Switch 1

Set OFF to read initial input from INITAUNIT (card reader).

Set ON to read initial input from INTBUNIT (tape unit 11).

Switch 2

Set OFF.

Switches 3 and 4

These alteration switches may be used to cause the typing of optional messages (see page 5).

2. Set accumulator overflow keys ON.

3. Set Priority Channel A and Priority Channel B dials.

a. Set right-hand dials to OFF.

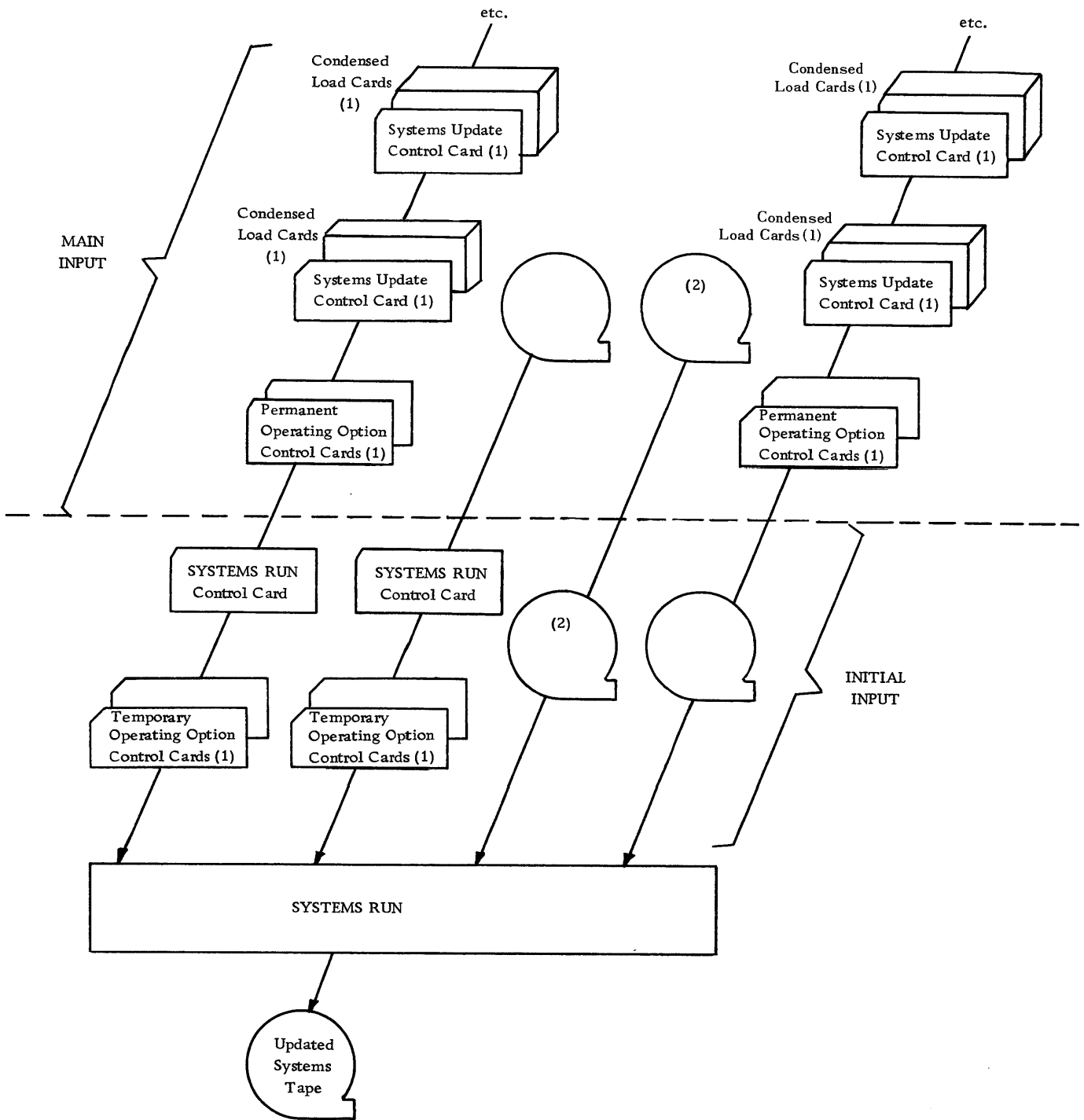
b. Set left-hand dials to N.

4. Press RUN.

5. Press COMPUTER RESET.

6. Press START.

Optional Configurations for Input Data



(1) This input will not necessarily appear during each run; see "Description."

(2) Main input and initial input may be on the same tape or on separate tapes mounted on separate units.

If warning messages were issued during the run, be careful to save the current systems tape.

Anticipated Halts:

Halt 0076 This halt indicates completion of the run. The new systems tape produced by this run will be on WORK-TAPEB (tape unit 12).

GENERATOR RUN

Description

The macro generator associated with a given macro statement produces the symbolic machine instructions that, collectively, perform the operation specified by the macro statement. New macro statements may be added to the source language at any time to fulfill a recurring need to state a particular operation. A macro generator program that will process the new statement must be written. It is then compiled, and the resultant machine-language instructions inserted in the systems tape. Both operations are accomplished by means of a Generator Run.

The macro generator program is written using the available Autocoder macro statements and symbolic machine language, and several special macros which perform functions common to all generators. Initially, a compilation of the macro generator program is performed during a Generator Run. A compiled generator program and a program listing are produced by this run.

The program listing which is produced must be carefully inspected for error messages and checked for accuracy. Modifications to the macro generator source-language input may be necessary; in this case, additional compilations must be made. When examination of the listing reveals that no further corrections are required, a final Generator Run is made. When the compilation is complete, the user may continue the run (in effect, starting a special systems run) to produce a new systems tape which will include the new macro generator. If a macro generator with the same name appears on the current systems tape, it will be replaced by the newly-compiled macro generator on the new systems tape.

The minimum input for a Generator Run consists of:

1. The GENERATOR RUN Control Card.
 2. The macro-generator source-language input.
- Prepare the RUN Control Card as follows:

Line	Label	Operation	OPERAND						Bcsi
			20	25	30	35	40	45	
01	GENERATOR	RUN	MNAME	P	S	SCANx	A		
02									

The operand portion of the GENERATOR RUN Control Card (columns 21-60) contains five different fields. Notice that these fields are separated by commas, and that no blanks are left between the fields. Beginning in column 21, as many columns (through 60) may be used as are needed for the fields. Unused columns after the last field used must be left blank.

The operand is punched as shown with the following codes describing the characteristics of the macro generator to be processed:

MNAME - The name of the macro generator. MNAME may be from one through five characters in length, and must begin with a letter. MNAME must be used as the label of the first executable instruction in the macro generator program.

P - The maximum number of parameters allowed in the operand of an input statement.

S - The maximum number of statements that might be generated for a single input statement.

SCANx - Indicates which scan in Phase I of Autocoder is to be used to process the input statements for this macro generator. The x is replaced by an A or a B. SCANA is to be used for fixed-format macro generators; SCANB is to be used for free-format macro generators. SCANB causes parameter records and an input text to be given to the macro generator. SCANA causes only parameter records to be given to the macro generator.

A - This field is called the Attributes Word. The Attributes Word has two purposes:

1. To permit Autocoder to distinguish functions from other macro generators. Function names and other macro generator names are all placed in the macro tables. Entries for function macro generators are two words long; the first word of each entry contains the name of the function macro generator, and the second word contains that function's Attributes Word. The two words must be in consecutive table locations. Entries for macro generators other than function

macro generators are one word long and consist solely of the macro generator name.

2. To provide information to macro generators which have functions as input parameters. The Attributes Word describes the characteristics of the result produced when the instructions generated by the function are executed at object time. For example, it may be necessary for the generator to see how the instructions generated by the function leave a result in the accumulators. It may even be necessary for the generator to generate instructions to make changes in the result e.g., shift the result in the accumulator to allow for more decimal places, or place the answer in a different accumulator. Since the function name is a parameter in a macro input statement, it has a parameter record. The Attributes Word is part of the function's parameter record. Thus, when a function parameter is found, a generator has the function's Attributes Word available. The various fields included in the Attributes Word are described below.

The Attributes Word is a ten-digit number without a sign.

<u>Digit</u>	<u>Explanation</u>
0	Attributes code:
0	The result has an unspecified or mixed alphabetic and numerical format.
1	The result is automatic-decimal and is right-justified in accumulators 1 and 2, coupled. Integers are included in this category.
2	The result is floating-decimal and is in accumulator 1.
3	The result is an address constant.
4	The result is alphameric and is left-justified in accumulator 1 or in accumulators 1 and 2 coupled.
5	The result is automatic-decimal in double-digit representation.
6	The result is floating-decimal in double-digit representation.
	(Note that codes 3, 4, 5, and 6 are <u>not</u> permitted if the function is to be used in ARITH statements.)
1-2	Number of integers in the result if the result is in the automatic-decimal mode. 00 if the result is in the floating-decimal mode.
3-4	Number of decimal places in the result if it is in the automatic-decimal mode. 00 if the result is in the floating-decimal mode.

5-7	Field size. The number of digits in the result if it is numerical, or the number of characters in the result if it is alphameric.
8	Always contains a zero.
9	Indicates to the macro generator with a function as an input parameter which of the four calling sequences to generate. The following codes are used: 9 for Type 1 8 for Type 2 7 for Type 3 6 for Type 4

Operating Procedure for a Generator Run

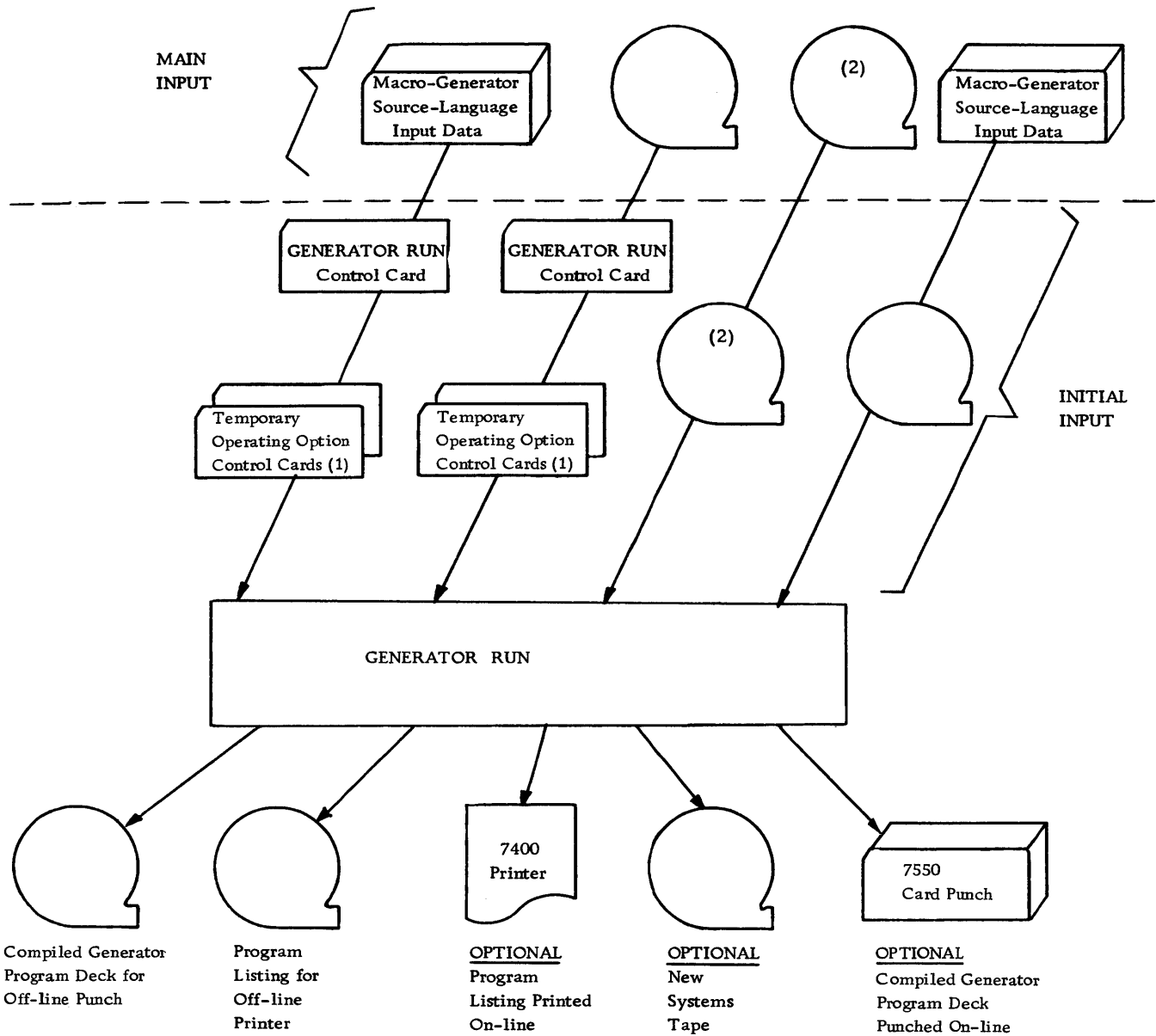
Ready the indicated input/output units as follows:

1. Mount the Compiler Systems Tape on SYSTEMTAPE (Tape unit 10).
Set the tape unit to high density.
2. If the initial input data is in punched card form:
 - a. Set card reader alteration switches to BBAA.
 - b. Insert the IBM 7500 Utility Panel.
 - c. Load initial input control cards in INITAUNIT (card reader).
 - d. Mount the main input data tape on MAINUNIT (tape unit 11).
3. If the initial input data is on tape:
Mount the tape containing initial and main input data on INITBUNIT (tape unit 11).
4. Mount work tapes on the following tape units indicated for a one-channel or a two-channel operation:

<u>Tape Unit</u>	<u>One Channel</u>	<u>Two Channels</u>
WORKTAPEB	tape unit 12	tape unit 12
WORKTAPEC	tape unit 13	tape unit 20
WORKTAPED	tape unit 14	tape unit 21
WORKTAPEE	tape unit 15	tape unit 22

5. If the program listing is to be printed on-line:
 - a. Set printer alteration switches to AAAA.
 - b. Insert the IBM 7400 Utility Panel.
6. If the object program deck is to be punched on line:
 - a. Set punch alteration switches to:
AAAA for a 12-punch in column 65.
ABAA for a 12-punch in column 79.
BAAA for a 12-punch in neither column.
 - b. Insert the 7550 Utility Panel.
 - c. Load punch hopper with cards (IBM electro C54549).

Optional Configurations for Input Data and Final Output



(1) This input is necessary only if the characteristics of the processor or object machines are to be altered for the current run.

(2) Main input and initial input may be on the same tape or on separate tapes mounted on separate units.

Store the following instructions manually:

<u>Location</u>	<u>Instruction</u>
0000	-8101010004
0001	-0100030010
0002	+5100110002

Set console controls for operation as follows:

1. Set alteration switches as follows:

Switch 1

Set OFF to read initial input from INITAUNIT (card reader).

Set ON to read initial input from INITBUNIT (tape unit 11).

Switch 2

Set OFF.

Switches 3 and 4

These alteration switches may be used to cause the typing of optional messages (see page 5).

2. Set accumulator overflow keys ON.
3. Set Priority Channel A and Priority Channel B dials.
 - a. Set right-hand dials to OFF.
 - b. Set left-hand dials to N.
4. Press RUN.
5. Press COMPUTER RESET.
6. Press START.

Anticipated Halts:

Halt 0003 This halt indicates that the main input tape on MAINUNIT (tape unit 11) has been read. This tape may be replaced by a work tape. See "Programmed Halts and Messages," page 30.

Halt 0067 This halt indicates that the compilation of the macro generator is complete. A message is typed identifying the tape units which contain the program listing and the compiled generator program deck. The newly-compiled macro generator may be entered on a new systems tape by pressing START.

Halt 0076 This halt indicates completion of the run. Remove and save the current systems tape on tape unit 10 and the new systems tape on the tape unit indicated by the typed message.

MULTIFILE RUN

Description

When two or more source programs are to be compiled, utilization of the Multifile Run by the operator will result in important savings in set-up and handling time. This run is designed to process, individually, several source programs read from the same input unit and to write all output on the same tape units (and punch or printer, if specified). The input programs may each be written using a different source language. The COBOL and Commercial Translator Systems do not, as yet, contain any provision for Multifile Runs.

As indicated in Table II, a minimum of seven tape units are required, the six required for a Compile Run and one for the program listing. As many as three additional tape units, or a maximum of ten units, may be utilized during a Multifile Run. The units for this run are assigned by means of control cards as follows:

1. LISTTAPE - one tape unit must always be available for the program listing. This tape will also be used for the object program deck if the MIXEDTAPE option (see page 26) is elected.
2. DECKTAPE - one additional unit may be used for the object program deck. The on-line punch must be designated by PUNCHDECK if an additional tape unit is not available or if the MIXEDTAPE option is not elected.
3. MAINUNIT - one additional unit may be used for main input if it is on tape. MAINUNIT must be the card reader if an additional tape unit is not available. WORKTAPA (tape 11) may not be used as MAINUNIT in a Multifile Run.
4. INITAUNIT - one additional unit may be used for initial input if it is not on the same tape as main input. If an additional tape unit is not available, the card reader must be used.

Note: On the original systems tape, the card reader is designated as INITAUNIT.

The minimum input for the Multifile Run consists of:

1. The MULTIFILE RUN Control Card followed by

2. Two or more sets, each containing in the order indicated:
 - a. a COMPILE RUN Control Card
 - b. the source-language input data
 - c. an END card to indicate termination of the source-language input data
 - d. a blank card (not required after the last program to be compiled)

With tape input, a tape mark must follow the last blank card only.

Prepare the MULTIFILE RUN Control Card as follows:

Line	Label	Operation	OPERAND					
3	56	1516	2021	25	30	35	40	45
01	MULTIFILE	RUN						
02								

For this run only, COMPILE RUN Control Cards are considered to be "main" input and, as such, are read from MAINUNIT. The MULTIFILE RUN Control Card will be the last (and perhaps only) input read from the "initial" input device.

One of the following COMPILE RUN Control Cards must be used for each program to be compiled:

Line	Label	Operation	OPERAND					
3	56	1516	2021	25	30	35	40	45
01	COMPILE	RUN	AUTOCODER					
02	COMPILE	RUN	FORTRAN					
03	COMPILE	RUN	REPORT GENERATOR					
04								

Line 01 shows the control card used when the input is in Autocoder language.

Line 02 shows the control card used for input that is in FORTRAN language.

The control card shown on Line 03 is used when the input is in the form of Report Program Generator specifications.

When an object program tape or the MIXEDTAPE option is used, the output for each program is followed by a tape mark to facilitate keeping the decks separate when they are punched off-line.

The number of source programs that may be compiled during a single Multifile Run may be limited by any of the following:

1. The number of source programs that can be written on one input reel of tape, if tape is used.
2. The number of output program listing records that can be written on one output reel of tape.
3. The number of output object program deck records that can be written on one output reel of tape, if tape is used.

Although all programs to be compiled in a Multifile Run need not be written using the same source language, all programs in the run must be compiled on a machine having the same operating options. The

use of temporary operating option control cards will affect the entire Multifile Run rather than the compilation of any one program.

Operating Procedure for a Multifile Run

Ready the indicated Input/Output units as follows:

1. Mount the Compiler Systems Tape on SYSTEMTAPE (tape unit 10). Set the tape unit to high density.
2. If the initial input data is in punched cards:
 - a. Set card reader alteration switches to BBAA.
 - b. Insert the IBM 7500 Utility Panel.
 - c. Load initial input control cards in INITAUNIT (card reader).
 - d. Mount the main input data tape on MAINUNIT ()*.
3. If the initial input data is on tape:
 - Mount the tape containing initial and main input data on MAINUNIT ()*
4. Mount work tapes on the tape units indicated for a one-channel or a two-channel operation:

<u>Tape Unit</u>	<u>One Channel</u>	<u>Two Channels</u>
WORKTAPEA	tape unit 11	tape unit 11
WORKTAPEB	tape unit 12	tape unit 12
WORKTAPEC	tape unit 13	tape unit 20
WORKTAPE D	tape unit 14	tape unit 21
WORKTAPEE	tape unit 15	tape unit 22
LISTTAPE	()*	()*

5. If the object program deck is to be written on tape, mount a work tape on DECKTAPE ()*. This tape will always be the same as that designated as LISTTAPE if the MIXEDTAPE option is elected.

6. If the object program deck is to be punched on-line:

- a. Set punch alteration switches to:
 - AAAA for a 12-punch in column 65.
 - ABAA for a 12-punch in column 79.
 - BAAA for a 12-punch in neither column.
- b. Insert the IBM 7550 Utility Panel.
- c. Load punch hopper with cards (IBM electro C54549).

7. If the program listing is to be printed on-line:

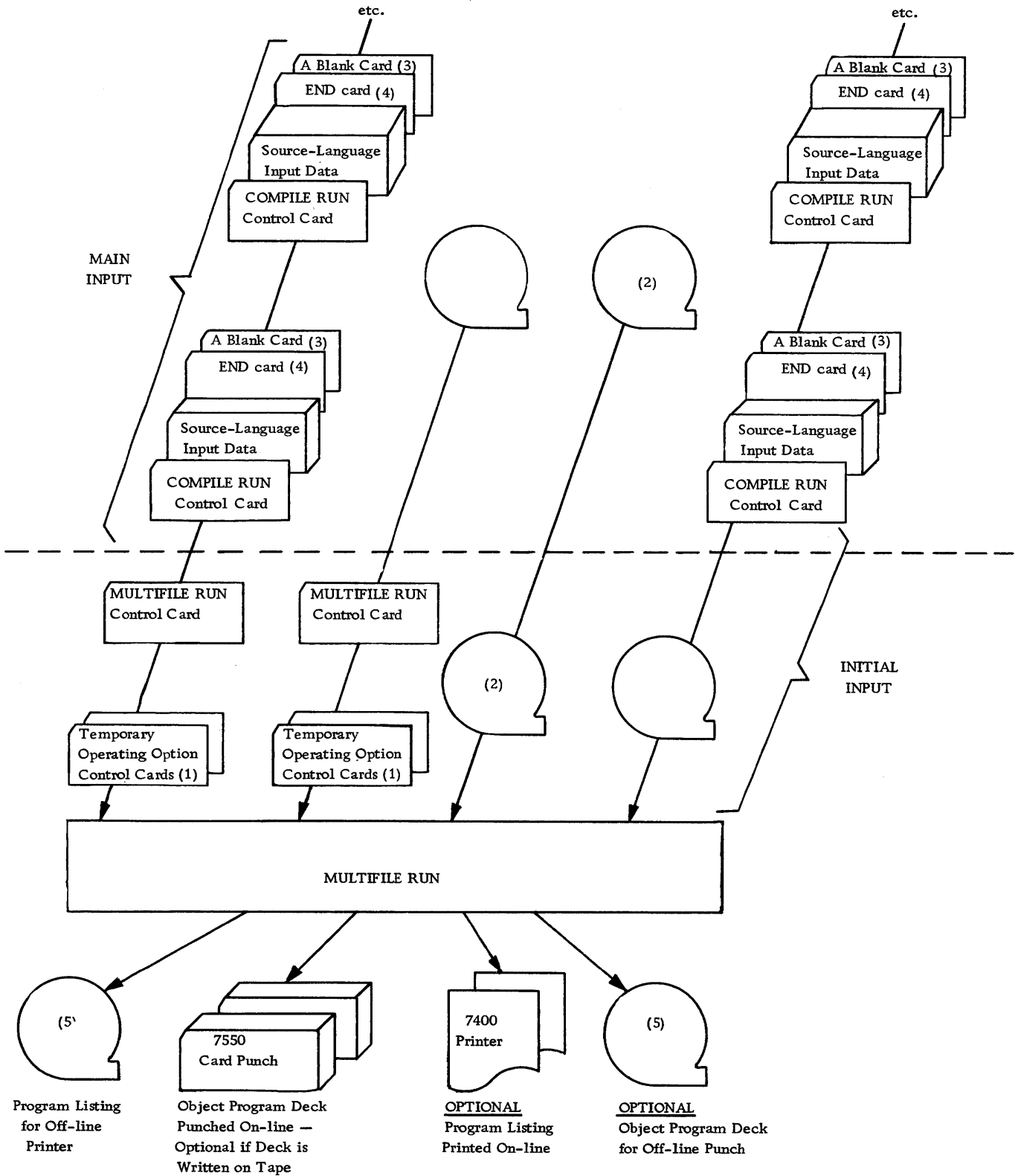
- a. Set printer alteration switches to AAAA.
- b. Insert the IBM 7400 Utility Panel.

Store the following instructions manually:

<u>Location</u>	<u>Instruction</u>
0000	-8101010004
0001	-0100030010
0002	+5100110002

*To be designated at each installation; see "Description," page 15.

Optional Configurations for Input Data and Final Output



- (1) This input is necessary only if the characteristics of the processor or object machines are to be altered for the current run.
- (2) Main input and initial input may be on the same tape or on separate tapes mounted on separate units.
- (3) This card is not necessary after the last program to be compiled.

- (4) This card is the appropriate end card for the source-program language.
- (5) Program listing and object program deck may be on the same tape or on separate tapes mounted on separate units.

Set console controls for operation as follows:

1. Set alteration switches as follows:

Switch 1

Set OFF to read initial input from
INITAUNIT ()*.

Set ON to read initial input from
INITBUNIT ()*.

Switch 2

Set OFF.

Switches 3 and 4

These alteration switches may be used
to cause the typing of optional messages
(see page 5).

2. Set accumulator overflow keys ON.

3. Set Priority Channel A and Priority Channel B dials.

- a. Set right-hand dials to OFF.
- b. Set left-hand dials to N.

4. Press RUN.

5. Press COMPUTER RESET.

6. Press START.

When the message xxTH PROGRAM COMPILED is typed, indicating that compilation of a source program has been completed, remove and identify any on-line output which may have been produced.

Anticipated Halts:

Halt 0076 - This halt indicates completion of the run. A message is typed identifying the tape units which contain the program listing and the object deck.

*To be designated at each installation; see "Description," page 15.

CONTROL CARDS: LISTING AND DESCRIPTION

The unique flexibility of the Compiler Systems Tape is accomplished by the use of control cards. These cards are punched according to the format of the 7070 Autocoder Coding Sheet, form X28-6417-2. However, columns 1 through 5 and 76 through 80 of these cards are never considered by the systems tape, so it is not necessary to include page number, line number or program identification. The operation to be performed is specified in columns 16 through 20 of the control card. This operation is qualified by information placed in columns 6 through 15, and, where necessary, by additional information which is punched beginning with column 21.

RUN CONTROL CARDS

A RUN Control Card is necessary for each use of the Compiler Systems Tape. In most cases, it will be the only control card used. The use of each card is explained in the operating instructions for its respective runs.

When the source input is written in Autocoder language, use the following COMPILE RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	COMPILE	RUN	AUTOCODER						
02									

When the source program is written in COBOL language, use the following COMPILE RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	COMPILE	RUN	COBOL						
02									

When the source program is written in Commercial Translator language, use the following COMPILE RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	COMPILE	RUN	COMMERCIAL TRANSLATOR						
02									

When the source program is written in FORTRAN language, use the following COMPILE RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	COMPILE	RUN	FORTRAN						
02									

When the source input is in the form of Report Program Generator specifications, use the following COMPILE RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	COMPILE	RUN	REPORT GENERATOR						
02									

If the systems tape is to be duplicated, use the following SYSTEMS RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	SYSTEMS	RUN	COPY						
02									

If the systems tape is to be updated, use the following SYSTEMS RUN Control Card:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	SYSTEMS	RUN							
02									

If the systems tape is to be updated by an official IBM patch or addition, a card of the following general format will be supplied by IBM:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	SYSTEMS	RUN	IBM	XXXXXX	YYYY	YYDD	DDDD		
02									

When a macro generator is to be compiled and (if desired) entered on a new systems tape, use a GENERATOR RUN Control Card punched as follows:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	GENERATOR	RUN	MNAME	P	S	S	CAN	X	A
02									

The operand is prepared by the author of the new macro generator. See page 12.

To allow two or more programs to be compiled in a single run, use a MULTIFILE RUN Control Card punched as follows:

Line	Label	Operation	OPERAND					Bas	
3	5/6	15/16	20/21	25	30	35	40	45	80
01	MULTIFILE	RUN							
02									

OPERATING OPTION CONTROL CARDS

The 7070 Series Compiler Systems Tape, as furnished by IBM, will include the permanent operating options listed on the coding sheet on page 20. These options may be changed by inserting the required operating option control card. When used before a RUN Control Card, they cause changes to be made for the current run only, i.e.,

their effect is temporary. When used after a SYSTEMS RUN Control Card, they cause permanent changes to be incorporated into the new systems tape. The line number corresponding to each operating

option listed on the coding sheet identifies each control card with its explanation on one of the following pages. These line numbers need not be punched in the control cards.

Line	Label	Operation	OPERAND											
			3	5	6	15	16	20	21	25	30	35	40	45
01	CHANNELS	EQUIP	2											
02	HIGHMEMORY	EQUIP	4	9	9	9								
03	PUNCHSYNC	EQUIP	1											
04	PRINTSYNC	EQUIP	2											
05	MAINUNIT	EQUIP	1	1										
06	INDENSITY	EQUIP	LOW											
07	INLABELS	EQUIP	NO											
08	WORKLABELS	EQUIP	LOW											
09	LISTTAPE	EQUIP	NONE											
10	DECKTAPE	EQUIP	NONE											
11	OUTDENSITY	EQUIP	LOW											
12	OUTLABELS	EQUIP	YES											
13	SEVENTHUNIT	EQUIP	2	5										
14	ADDSTORAGE	OBJCT	NO											
15	MAXMEMORY	OBJCT	2	9	9	9	9							
16	MINMEMORY	OBJCT	0	0	0	0	0							
17	LOADPROG	OBJCT	YES											
18	FLOATDECML	OBJCT	NO											
19	STARTLOCN	OBJCT	0	3	2	5								
20	PRINTLINES	OBJCT	5	0										
21	LOADMODE	OBJCT	ALPHA											
22	LOADMODEF	OBJCT	NUM											
23	LOADPUNCH	OBJCT	7	9										
24	LOADPUNCHF	OBJCT	7	9										
25	PUNCHDECK	OBJCT	NO											
26	PRINTLIST	OBJCT	NO											
27	LISTBLOCK	OBJCT	1											
28	ERRORPRINT	OBJCT	NO											
29	MIXEDTAPE	OBJCT	NO											
30	XRESERVE	OBJCT	(SEE NOTE)											
31	XRELEASE	OBJCT	1-96											
32	SRESERVE	OBJCT	(SEE NOTE)											
33	SRELEASE	OBJCT	1-30											
34	INITAUNIT	EQUIP	1											
35	INITBUNIT	EQUIP	1	1										

Note: XRESERVE and SRESERVE are not included on the systems tape as furnished by IBM since all index words and electronic switches are made available to the user.

Processor Machine

Control cards which indicate either the characteristics of the processor machine on which the Compiler Systems Tape will be run or various options concerning the input to a given run are punched as follows:

1. The tape units needed to process a program using the systems tape may all be on channel 1 or may be divided between channels 1 and 2. The number of tape channels to be used is specified in a control card as follows:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	CHANNELS	EQUIPN						
0.2								

The value of N in the operand may be either 1 or 2. If all tape units are connected to channel 1, the value of N must be 1. If N is 2, there must be a minimum of three tape units connected to both channel 1 and channel 2.

2. The machine on which the systems tape will be run may have more than 5,000 words available for processing. The highest storage location is specified in a control card as follows:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	HIGHMEMORY	EQUIPN	NNNN					
0.2								

The value of NNNN may be any number from 4999 through 9989 which specifies the highest storage location available to the processor machine. If the processor machine has the optional additional core storage feature, the machine must be set to operate in the normal storage mode by setting the Additional Storage switch OFF.

3. If an on-line IBM 7550 Card Punch is available, specify the unit record synchronizer in a control card as follows:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	PUNCHSYNC	EQUIPN						
0.2								

The value of N may be 1, 2, or 3 to specify the number of the output synchronizer to which the punch is attached. This synchronizer must be specified if the option of on-line punching of the condensed object program deck is used.

4. If an on-line IBM 7400 Printer is available, specify the unit record synchronizer in a control card as follows:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	PRINTSYNC	EQUIPN						
0.2								

The value of N may be 1, 2, or 3 to specify the number of the output synchronizer to which the printer is attached. This synchronizer must be specified if the option of on-line printing of the program listing is used.

5. To establish the input device to be used to read all cards which follow the RUN Control Card for a specific program, use one of the following control cards:

A. For card input.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	MAINUNIT	EQUIPN						
0.2								

The value of N may be 1, 2, or 3 to specify the number of the input synchronizer to which the IBM 7500 Card Reader is attached.

B. For tape input.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	MAINUNIT	EQUIPCU						
0.2								

The operand specifies the tape channel (C) and the tape unit (U) on which the card image input tape is mounted. The unit used for input may be:

1. WORKTAPEA (channel 1, unit 1) except for Multifile Runs.
2. Any unique tape other than one of the required six; see Table I, page 4.

6. If an input tape is used, the tape file may be in either high or low density; the density is specified by using one of the following control cards:

A. For input files written in low density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	LDENSITY	EQUIPLW						
0.2								

B. For input files written in high density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	LDENSITY	EQUIPHIGH						
0.2								

7. Labels are optional on an input tape when it is not utilized as a work tape or when labels are not used on the work tapes. The use of labels on input tapes, and the density at which the labels are written, are specified by using one of the following control cards:

A. The input tape does not have labels.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	INLABELS	EQUIRNO						
0.2								

B. The input tape has labels written in low density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	IN LABELS	EQUIP	LOW					
0.2								

C. The input tape has labels written in high density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	IN LABELS	EQUIP	HIGH					
0.2								

An input tape may also be used later in the same run as a work tape. If labels are specified for the work tapes, the input tape must have standard Input/Output Control System labels of the same density. However, if the input tape file is in low density, the input labels must also be in low density. In this instance, therefore, the labels of the work tapes must be in low density.

8. The use of standard Input/Output Control System tape labels on the work tapes is optional. If used, work labels must be written in low density when either an input tape is to be used as a work tape and the input file is written in low density, or the file of a labeled output tape is written in low density. The use of labels on work tapes and the density at which the labels are to be processed are specified by using one of the following control cards:

A. The work tapes do not have labels.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	WORK LABELS	EQUIP	NONE					
0.2								

B. The work tapes have labels written in low density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	WORK LABELS	EQUIP	LOW					
0.2								

C. The work tapes have labels written in high density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	WORK LABELS	EQUIP	HIGH					
0.2								

9. For a Multifile Run, at least one tape unit (in addition to the six required for a Compile Run) must be available for the program listing. The tape unit to be used for the program listing is specified by using the following control card:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	LIST TAPE	EQUIP	CU					
0.2								

In the operand, C is the channel number and U is the tape unit number of any tape unit other than one of the required six; see "Operating Procedure for a Multifile Run," page 16. During initial label processing, this tape is considered to be a work tape.

If an additional tape unit for the program listing is not available, use the following control card:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	LIST TAPE	EQUIP	NONE					
0.2								

When NONE is specified in the operand, A Multifile Run is not possible.

10. For a Multifile Run in which a condensed object program deck is to be written on tape for subsequent off-line punching, a tape unit must be available in addition to the minimum of seven required for a Multifile Run. The tape unit to be used for the object program deck is specified by using the following control card:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	DECK TAPE	EQUIP	CU					
0.2								

C is the channel number and U is the tape unit number of any tape unit other than one of the required seven. During initial label processing, the additional tape is considered to be a work tape.

If an additional tape unit for the object program deck is not available, use the following control card:

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	DECK TAPE	EQUIP	NONE					
0.2								

When NONE is specified in the operand, the object program deck must be obtained by on-line punching.

11. The output tape files may be written in either high or low density; the density is specified by using one of the following control cards:

A. For output files to be written in low density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	OUT DENSITY	EQUIP	LOW					
0.2								

B. For output files to be written in high density.

Line	Label	Operation	OPERAND					
3	5/6	15/16	20/21	25	30	35	40	45
0.1	OUT DENSITY	EQUIP	HIGH					
0.2								

12. The use of standard Input/Output Control System labels on the final output tapes is optional.

The use of labels on the output tapes is specified by one of the following control cards:

A. The output tapes are not to have labels.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	OUT LABELS	EQUIP NO						
0.2								

B. The output tapes are to have labels.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	OUT LABELS	EQUIP YES						
0.2								

If work labels are not used, the density of the output labels will be the same as that of the output file. If work labels are used, the density of the output labels will be the same as that of the work labels. However, when output labels are used, they must be in low density if the output file is in low density. In this instance, if the work tapes are labeled, their density must also be low.

13. To specify the additional (i. e., the seventh) tape unit needed by COBOL and Commercial Translator, a card of the following form is used:

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	SEVENTH UNIT	EQUIP CU						
0.2								

CU specifies the channel and unit number of the seventh tape. The systems tape, as furnished by IBM, specifies tape unit 25 as the permanent operating option for the seventh unit.

Object Machine

Certain characteristics of the object machine on which the assembled program or resultant systems tape will be run, as well as various options concerning the output of a given run, must be specified to the Compiler Systems Tape. The control cards for these purposes are punched as follows:

14. Additional core storage is an optional feature on the IBM 7072 and 7074. Systems with this feature can operate either in the additional storage mode or in the standard mode. For a particular compilation, use one of the following control cards to indicate the mode in which the object machine will operate:

A. The object machine is to be operated in the additional storage mode.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	ADDSTORAGE	OBJECT	YES					
0.2								

B. The object machine is to be operated in the standard mode.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	ADDSTORAGE	OBJECT	NO					
0.2								

The Compiler Systems Tape, as furnished by IBM, has ADDSTORAGE set to NO.

15. The highest storage location that can be assigned when compiling an object program must be specified; the following control card is used to specify this location:

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	MAXMEMORY	OBJECT	NNNNN					
0.2								
0.3								

The value of NNNNN in the operand must be less than, or equal to, the maximum size of the system. In no case can it exceed 29999.

Note: Any control card dealing with this option will be disregarded by the FORTRAN compiler, which sets MAXMEMORY at 9989. However, programs compiled by the FORTRAN compiler can be run on a 5K 7070 Series machine. The FORTRAN Relocatable Loader will position these programs properly in storage.

16. The lowest storage location that can be assigned when compiling an object program must be specified; the following control card is used to specify this location:

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	MINMEMORY	OBJECT	NNNNN					
0.2								

The value of NNNNN in the operand must be less than, or equal to, the maximum size of the system. In no case can it exceed 29999.

Note: Any control card dealing with this option will be disregarded by the FORTRAN compiler, which sets MINMEMORY at 0325.

17. The output of a Compile Run may include the five-card IBM 7070/7074 Condensed Card Load Program. The following control cards specify whether the load program is to be produced:

A. The load program is to be included.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	LOADPROG	OBJECT	YES					
0.2								

B. The load program is not to be included.

Line	Label	Operation	OPERAND					
3	5-6	15-16	20-21	25	30	35	40	45
0.1	LOADPROG	OBJECT	NO					
0.2								

Note: Any control card dealing with this option will be disregarded by the FORTRAN compiler, which sets LOADPROG to NO.

18. The ability to do floating-decimal operations is an optional feature available for the 7070 or 7074. The following control cards specify whether this optional feature is installed on the object machine:

A. The object machine has the floating-decimal optional feature.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	FLOATDECMI		OBJCTYES					
0 2								

B. The object machine does not have the floating-decimal optional feature.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	FLOATDECMI		OBJCTNO					
0 2								

19. The address that will be considered by the compiler when assigning the initial storage location may be altered by using the following control card:

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	STARTLOCN		OBJCTNNNN					
0 2								

The value of NNNN in the operand must be less than, or equal to, the maximum size of the system. In no case can it exceed 29999.

20. The number of lines per page, including header lines, desired when printing the program listing may be specified by using the following control card:

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	PRINTLINES		OBJCTNN					
0 2								

The value of NN in the operand may be any number from 01 through 99 to specify the number of lines per page to be printed. If NN equals 99, a skip to a new page will occur after line 49 is printed. The new page will contain lines 50 through 99.

21. The output of a Compile Run will include a condensed card image tape which may be produced in alphanumeric or numerical form. The form is specified by using one of the following control cards when the source program language is not FORTRAN:

A. The source program language is not FORTRAN, and the tape is to consist of card image records, each containing eight numerical words preceded by a delta and no load card indicator.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	LOADMODE		OBJCTNUM					
0 2								

The numerical card image tape may be loaded as direct input when the object program is run. An object program deck cannot be punched off-line from the numerical card image tape unless the card punch is equipped with a delta suppressor or unless a 1401 is programmed to ignore the delta.

B. The source program language is not FORTRAN, and the tape is to consist of card image records, each containing 80 alphanumeric characters (not preceded by a delta) and an optional load card indicator.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	LOADMODE		OBJCTALPHA					
0 2								

An object program deck can be punched off-line from the alphanumeric card image tape.

22. The output of a Compile Run will include a condensed card image tape which may be produced in alphanumeric or numerical form. The form is specified by using one of the following control cards when the source program language is FORTRAN:

A. The source program language is FORTRAN, and the tape is to consist of card image records, each containing eight numerical words preceded by a delta and no load card indicator.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	LOADMODE		OBJCTNUM					
0 2								

The numerical card image tape may be loaded as direct input when the object program is run. An object program deck cannot be punched off-line from the numerical card image tape unless the card punch is equipped with a delta suppressor or unless a 1401 is programmed to ignore the delta.

B. The source program language is FORTRAN, and the tape is to consist of card image records, each containing 80 alphanumeric characters (not preceded by a delta) and an optional load card indicator.

Line	Label	Operation	OPERAND					
3	36	1516	2021	25	30	35	40	45
0 1	LOADMODE		OBJCTALPHA					
0 2								

An object program deck can be punched off-line from the alphanumeric card image tape.

23. A load card indicator (12-punch) can be included in each record of the alphameric condensed card image tape produced by a compilation. When the source program language is not FORTRAN, the following control cards specify whether or not a load card indicator is to be included.

- A. The source program language is not FORTRAN, and a load card indicator is to be included.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	LOADPUNCH	OBJECTNN						
02								

The value of NN in the operand may be any value from 01 through 80 to specify the column in each record in which the load card indicator is to appear.

- B. The source program language is not FORTRAN, and a load card indicator is not to be included.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	LOADPUNCH	OBJECTNONE						
02								

24. A load card indicator (12-punch) can be included in each record of the alphameric condensed card image tape produced by a compilation. When the source program language is FORTRAN, the following control cards specify whether or not a load card indicator is to be included.

- A. The source program language is FORTRAN, and a load card indicator is to be included.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	LOADPUNCH	OBJECTNN						
02								

The value of NN in the operand may be any value from 01 through 80 to specify the column in each record in which the load card indicator is to appear.

- B. The source program language is FORTRAN, and a load card indicator is not to be included.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	LOADPUNCH	OBJECTNONE						
02								

25. The condensed program deck produced during a Compile Run or a Generator Run may be punched on-line on an IBM 7550 Card Punch, in addition to being written on tape for off-line punching.

The following control cards specify whether on-line punching is desired:

- A. The program deck is to be punched on-line.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	PUNCHDECK	OBJECTYES						
02								

When on-line punching is specified, the output synchronizer (PUNCHSYNC) to which the IBM 7550 Card Punch is connected must be available to the compiler.

- B. The program deck is not to be punched on-line.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	PUNCHDECK	OBJECTNO						
02								

26. The program listing produced during a Compile Run or a Generator Run may be printed on-line on an IBM 7400 Printer, in addition to being written on tape for off-line printing. The following control cards specify whether on-line printing is desired:

- A. The program listing is to be printed on-line.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	PRINTLIST	OBJECTYES						
02								

When on-line printing is specified, the output synchronizer (PRINTSYNC) to which the IBM 7400 Printer is connected must be available to the compiler.

- B. The program listing is not to be printed on-line.

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	PRINTLIST	OBJECTNO						
02								

27. Records on the output listing tape may be blocked. The blocking factor is specified in a control card as follows:

Line	Label	Operation	OPERAND					
3	56	15/16	20/21	25	30	35	40	45
01	LISTBLOCK	OBJECTN						
02								

The value of N may be from 1 through 5. If N is 1, the length of the tape records will be 120 characters without record marks. If N equals 2, 3, 4, or 5, all records will contain record marks and will vary in size (in increments of 5 characters) from 25 to 125 characters. These records may be printed on an IBM 1401 by using the Multiple Utility Program. The Compiler Systems Tape, as furnished by IBM, has N equal to 1.

When the records are blocked, their lengths (including record marks) are as follows:

	<u>Number of Characters</u>	
	<u>Heading</u>	<u>Listing</u>
Label	80	-
Program Listing	125	125
Origin Counter Listing	95	95
Electronic Switch Availability Table	25-35	85
Index Word Availability Table	25-35	85
Cross Reference Listing	40-70	125
Electronic Switch Usage Listing	125	125
Index Word Usage Listing	125	125
Halt Listing	125	125
Message Listing	125	125

Note: When the Compiler Systems Tape is to be run on a 5K machine, setting LISTBLOCK to 3, 4, or 5 will cause the Input/Output Control System to use one storage area for each tape file, thus increasing compilation time.

28. Diagnostic messages produced by COBOL and Commercial Translator are written on tape. Users may indicate, by means of one of the following option cards, whether or not such messages are also to be listed by the on-line printer.

A. Messages are to be printed.

Line	Label	Operation	OPERAND
5	5/6	15/16	20/21 25 30 35 40 45
0.1	ERRORPRINT	OBJCT	YES
0.2			

B. Messages are not to be printed.

Line	Label	Operation	OPERAND
5	5/6	15/16	20/21 25 30 35 40 45
0.1	ERRORPRINT	OBJCT	NO
0.2			

29. The program listing and condensed object program deck produced by a Compile Run or a Multifile Run may both be written on the same tape. The condensed card images (numeric and/or alphanumeric) will be interspersed on this tape and must subsequently be separated by an appropriate sort program. The selection of this option does not affect the results produced by the PUNCHDECK, PRINTLIST, and LISTBLOCK options. One of the following control cards is used to specify whether this combined tape is to be produced:

A. The program listing and object program deck are to be written on the same tape.

Line	Label	Operation	OPERAND
5	5/6	15/16	20/21 25 30 35 40 45
0.1	MIXEDTAPE	OBJCT	YES
0.2			

For a Multifile Run, this tape will always be the one mounted on the tape unit designated as LISTTAPE. For a Compile Run, this tape will be on the tape unit, if any, designated as LISTTAPE; otherwise it will be one of the work tapes.

B. The program listing and object program deck are to be written on separate tapes.

Line	Label	Operation	OPERAND
5	5/6	15/16	20/21 25 30 35 40 45
0.1	MIXEDTAPE	OBJCT	NO
0.2			

The use of the MIXEDTAPE option reduces the number of tape units required for Multifile Runs on a tape-oriented system from eight to seven. This option cannot be used for Generator Runs; the MIXEDTAPE Control Card will be ignored if it appears as part of the input for such a run.

Permanent Operating Option Changes

The following control cards are used only for changing permanent operating options on the Compiler Systems Tape. They must follow a SYSTEMS RUN Control Card:

30. A table of available index words is used during compilation for the assignment of symbolic index word names. The initial setting of this table can be altered when a new systems tape is created by using the following control card:

Line	Label	Operation	OPERAND
5	5/6	15/16	20/21 25 30 35 40 45
0.1	XRESERVE	OBJCTN	1, N2- N3, N4, etc.
0.2			

The operand contains the one-digit or two-digit number of the index word(s) (N_1 , N_2 , etc.) to be reserved. Individual index words or sequences of index words must be separated by commas. Sequences of index words may be reserved by specifying the first and last index words of a sequence, and separating them with a hyphen. For example, $N_2 - N_3$ refers to all index words from N_2 through N_3 inclusive.

Note: Index words 97, 98, and 99 are not to be used in the operand of the XRESERVE Control Card. These index words are not available because of their special function in table lookup and priority processing operations. If 97, 98, or 99 appears in the operand, the entire control card will be considered invalid and will be ignored.

31. Index words that have been reserved previously may be made available for assignment by using the following control card:

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	XRELEASE	OBJECTN	1, N2, N3, N4, etc.			
02						

The operand contains the one-digit or two-digit number of the index word(s) (N_1 , N_2 , etc.) to be made available to the compiler. Individual index words or sequences of index words must be separated by commas. Sequences of index words are specified by separating the numbers of the first and last index words of a sequence with a hyphen. For example, N_1-N_2 refers to all index words from N_1 through N_2 inclusive.

Note: Index words 97, 98, and 99 are not to be used in the operand of the XRELEASE Control Card. These index words are not available because of their special function in table lookup and priority processing operations. If 97, 98, or 99 appears in the operand, the entire control card will be considered invalid and will be ignored.

32. A table of electronic switches is used during compilation for the assignment of symbolic electronic switch names. The initial setting of this table can be altered when a new systems tape is created by using the following control card:

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	SRESERVE	OBJECTN	1, N2, N3 - N4, etc.			
02						

The operand contains the one-digit or two-digit number of the electronic switch(es) (N_1 , N_2 , etc.) to be reserved. Individual electronic switches or sequences of switches must be separated by commas. Sequences of electronic switches may be reserved by separating the numbers of the first and last switches of a sequence with a hyphen. For example, $N_3 - N_4$ refers to all electronic switches from N_3 through N_4 inclusive.

33. Electronic switches that have been reserved previously may be made available for assignment by using the following control card:

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	SRELEASE	OBJECTN	1, N2 - N3, N4, etc.			
02						

The operand contains the one-digit or two-digit number of the electronic switch(es) (N_1 , N_2 , etc.) to be made available to the compiler. Individual electronic switches or sequences of switches must be separated by commas. Sequences of electronic switches are specified by separating the first and

the last switches of a sequence with a hyphen. For example, $N_2 - N_3$ refers to all electronic switches from N_2 through N_3 inclusive.

34. To specify the initial input device to be used (to read all control cards up to and including the RUN Control Card) when alteration switch 1 is OFF, use one of the following control cards.

A. A card reader is to be used for initial input.

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	INITAUNIT	EQUIPN				
02						

The value of N in the operand specifies the number of the input synchronizer to which the card reader is connected.

B. A tape unit is to be used for initial input.

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	INITAUNIT	EQUIFCU				
02						

The operand specifies the tape channel (C) and the tape unit (U) on which the card image initial input tape is mounted. This unit may be:

1. Any tape unit other than one of the required six.
2. The same tape unit (if any) specified as MAINUNIT.

35. To specify the initial input device to be used (to read all control cards up to and including the RUN Control Card) when alteration switch 1 is ON, use one of the following control cards:

A. A card reader is to be used for initial input.

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	INITBUNIT	EQUIPN				
02						

The value of N in the operand specifies the number of the input synchronizer to which the card reader is connected.

B. A tape unit is to be used for initial input.

Line	Label	Operation	OPERAND			
3	5/6	15/16	20/21	25	30	35 40 45
01	INITBUNIT	EQUIFCU				
02						

The operand specifies the tape channel (C) and the tape unit (U) on which the card image initial input tape is mounted. This unit may be:

1. Any tape unit other than one of the required six.

2. The same tape (if any) specified as MAINUNIT.

2. To modify an existing coding block, use the following control card:

SYSTEMS UPDATE CONTROL CARDS

The following cards are valid only if they follow a SYSTEMS RUN Control Card or any subsequent operating option control cards. The actual ten-character name of the coding block to be affected must appear in columns 6 through 15. These cards must be arranged so that the ten-character names appear in ascending collating order. For additional restrictions, see page 9.

1. To delete the indicated coding block, use the following control card:

Line	Label	Operation	OPERAND
3	5	15	20
5	15	20	25
30	35	40	45
0 1	XXXXXXXXXX	DROP	
0 2			

The name of the coding block (Xs above) to be deleted is specified in the Label columns. If a coding block with this name does not appear on the old systems tape, this DROP Control Card will be ignored.

Line	Label	Operation	OPERAND
3	5	15	20
5	15	20	25
30	35	40	45
0 1	XXXXXXXXXX	PATCH	
0 2			

The name of the coding block (Xs above) to be modified is specified in the Label columns. The PATCH Control Card must be followed by condensed load cards which update the existing coding block by replacements or additions.

3. To enter a new coding block on the updated systems tape, use the following control card:

Line	Label	Operation	OPERAND
3	5	15	20
5	15	20	25
30	35	40	45
0 1	XXXXXXXXXX	ENTER	
0 2			

The name of the coding block (Xs above) to be entered is specified in the Label columns. The ENTER Control Card must be followed by condensed load cards to be used in the construction of the new coding block.

OPERATION AND INSTALLATION SUGGESTIONS

A significant feature of the 7070 Series Compiler Systems Tape is the facility with which it can be adapted to the particular needs of the user. Control cards describing the operating options which best suit the machine configurations and requirements at a given installation can be entered temporarily at the time that a run is made or may be included as a permanent part of the systems tape. The latter procedure is strongly recommended in order to simplify the preparation of input and to reduce the opportunity for input errors. The versatility of the systems tape can also be increased by the user, through the addition of new macro generators designed to cover unique programming operations.

It is recommended that one person at each installation be assigned the responsibility of authorizing and documenting all revisions. Thus, adequate control may be maintained over the contents of the systems tape.

A Systems Run is the initial run which should be made when the systems tape is received from IBM. First, the permanent operating options furnished on the tape (see page 20) should be carefully evaluated. If they conform to what is desired by the user, a duplicate of the systems tape should be produced by using a SYSTEMS RUN COPY Control Card as input. The furnished tape can be filed as a master and the new tape put to active use. If the user finds that modifications to the furnished operating options (e. g., item 6 below) are desirable for more effective utilization of equipment, a new systems tape should be produced. In this case the input will include a SYSTEMS RUN Control Card, followed by appropriate operating option control cards.

During a Systems Run, the typewritten log will include a list of any changes that are made. This log, properly identified as to the tape to which it applies, should be filed and kept current whenever a new systems tape is produced so that the latest status of a particular tape is always known.

The operating procedures described in this manual are based on the operating options in the original systems tape. They may be used as a guide in the preparation of instructions when these operating options have been modified. This is particularly true of the instructions for a Multifile Run since the additional tape(s) required must be designated by the user. Care must be taken to insure that the operating instructions are revised, if necessary, when a new systems tape is produced.

The operating option control cards that are used to designate the initial input devices (see "Permanent Operating Option Changes," page 26) may

appear following a Systems Run Control Card only, thus becoming a permanent part of a new systems tape; there is no way of temporarily changing the initial input units at the start of a run. Therefore, when the first run is made the required control cards must be read in either from a card reader on input synchronizer 1 or from tape unit 11. In addition, for this first run, appropriate operating option control cards must be placed before the RUN Control Card if the operating options at the user's installation differ from the ones listed below.

1. MAINUNIT - tape 11
2. INLABELS - NO
3. INDENSITY - LOW

While experience will indicate which operating procedures are most suitable for each installation, the suggestions which follow should serve as a guide when the necessary standard practices are being established by the user.

1. Errors due to incorrect control cards can be avoided by having the operating option and RUN Control Cards, where practical, pre-punched and readily accessible for the use of the machine operators and for those who may be preparing input for a run.

2. Provisions should be made to indicate to the operator the presence of any operating option control cards which will cause departures from standard procedures. This is particularly true when initial input is to be on tape since the control cards are not readily available for examination by the operator.

3. A run will be discontinued if an end-of-reel reflective spot is sensed on a work tape. To avoid this, the use of full length tape is recommended.

4. The use of tape labels is optional, but, when used, they must be consistent with standard Input/Output Control System usage.

5. Designation of improper operating options during a Systems Run will cause warning messages to be produced as the erroneous options are included on the new tape. In this case, it is particularly important to save the old systems tape for use as input to a later Systems Run with any necessary corrected control cards.

6. When more than six tape units are available, one should be designated as LISTTAPE so that Multifile Runs will be possible. If another tape unit is designated as DECKTAPE and no on-line output is specified, the speed at which output is produced during a Multifile Run may be increased.

7. For efficient computer operation, the use of on-line output should be held to a minimum.

PROGRAMMED HALTS AND MESSAGES

All Compiler Systems Tape programmed halts are Halt and Proceed (HP) instructions. The halts are listed according to the halt number, i e., digit positions six through nine of the program typeout. Thus, the typeout

1654 -0000001234

would refer to Halt 1234. If an associated message is typed, the text appears in capital letters to the right of the number. Each halt number is accompanied by an explanation and the action to be taken by the operator. If the halt is unique to a section, a run, or a programming system, this information appears at the beginning of the explanation, e.g., Compile Run, Phase 1, Autocoder.

A total of 100 read attempts is recommended for all halts where pressing START initiates nine attempts to reread.

Note: The FORTRAN Relocatable Loader programmed halts and messages are listed on page 47.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0001	YOUR SYSTEM TAPE IS VERSION xxxxx, THIS SYSTEMS RUN IS FOR VERSION yyyyy. DISCONTINUE RUN. <u>Explanation:</u> A Systems Run is being performed to effect IBM-originated changes and/or additions, but the Compiler Systems Tape version for which the modifications are intended is not the version of the Compiler Systems Tape being used in the Systems Run. <u>Action:</u> Discontinue the run.
0002	YOUR SYSTEMS TAPE CHANGE LEVEL IS xxxxx, THIS SYSTEMS RUN IS FOR CHANGE LEVEL yyyyy. <u>Explanation:</u> A Systems Run is being performed to effect IBM-originated changes and/or additions, but the Compiler Systems Tape change level for which the modifications are intended is not the change level of the Compiler Systems Tape being used in the Systems Run. Previous IBM-originated changes have not yet been made, or they have been made, but not in the order in which they were supplied. <u>Action:</u> It is recommended that this Systems Run be delayed until earlier changes have been made. The run may be continued, however, by pressing START.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0003	REPLACE TPxx TO SAVE SOURCE INPUT <u>Explanation:</u> The source input for this run has been read, and the tape unit(xx) used for mounting it is now needed as a work tape unit. <u>Action:</u> Replace the source input tape (if it is to be saved) with a work tape, and then press START. If the source input is not to be saved, merely press START. In either case, the tape reel mounted on this tape unit when the Start key is pressed must be identical to other work tapes as to presence or absence of tape labels and as to label density if labels are present.
0004	END OF FILE - NO RUN CARD DISCONTINUE RUN <u>Explanation:</u> The input device being used for reading initial input has reached end of file (no cards in card reader, or tape mark on tape) before a RUN Control Card is encountered. This situation is beyond the control of the Compiler System. <u>Action:</u> Discontinue the run, prepare the initial input properly, and then restart the run from the beginning.
0005	RUN CARD MISSING DISCONTINUE RUN <u>Explanation:</u> Multifile Run. The next source program to be compiled is not prefaced by a RUN Control Card. This situation is beyond the control of the Compiler System. <u>Action:</u> Discontinue the Multifile Run at this point. Listings and program decks of source programs (if any) which have been compiled prior to this halt should be removed from the multifile output units.
0006	xxxxxxxxxxxxxxxxxxxxxxxxx INVALID DISCONTINUE RUN <u>Explanation:</u> A RUN Control Card, identified by Xs above, is not valid. Either it does not conform to any possible run

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
	card format, or it follows a MULTIFILE RUN Control Card, but is not a COM-PILE RUN Control Card. This situation is beyond the control of the Compiler System.
	<u>Action:</u> Discontinue the run at this point. If this is a Multifile Run, listings and program decks of source programs (if any) that have been compiled prior to this halt should be removed from the multifile output units.
0007	INPUTS SPECIFY HIGH LABEL, LOW FILE INITIAL INPUT UNIT A IS WORK TAPE xx. INITIAL INPUT UNIT B IS WORK TAPE xx DISCONTINUE RUN
	<u>Explanation:</u> Any run. Before any initial input is read, a check is made to determine the correctness of the pertinent operating options. If not correct, the corresponding lines above are typed. The second or third message will occur when the selected initial input unit is found to be the same as a work tape (xx) other than MAINUNIT. Since current operating option control cards have not yet been read (and could not change the initial input device currently used), this halt indicates an improperly prepared systems type, i. e., incorrect operating option control cards followed the RUN Control Card in the Systems Run which created this systems tape (similar messages were produced at that time; the run should have been repeated with correct control cards).
	<u>Action:</u> If only the second or third message appears before the "DISCONTINUE RUN" message, start the run from the beginning, using the alternate initial input unit. If this unit is not available (or if it too is incorrectly designated), create a new systems tape, supplying correct operating option control cards. Note that the current systems tape will not be usable as input to a Systems Run.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0008	INITIAL AND MAIN INPUT ALLOCATION IMPROPER. MAIN INPUT UNIT IS WORD TAPE xx. INPUTS SPECIFY HIGH LABEL, LOW FILE. DISCONTINUE RUN.
	<u>Explanation:</u> Any run. After the RUN Control Card and any preceding operating option control cards are read, the correctness of the pertinent operating options is checked. If they are not correct, one or more of the first three lines will be typed. The first message appears when the processor finds that the initial input unit is a tape other than either MAINUNIT or a unique non-work tape. The second message appears when the processor finds that MAINUNIT is the same as a work tape other than WORKTAPEA (tape 11). These errors may result from: 1. Improper operating option cards preceding the RUN Control Card in this run. 2. Improper operating option control cards following the RUN Control Card in the Systems Run that created this systems tape (similar messages were produced at that time; the run should have been repeated with corrected control cards). 3. A combination of both of the above conditions.
	<u>Action:</u> Supply correct operating option control cards, and restart the run from the beginning. If necessary, create a new systems tape, supplying correct operating option control cards.
0009	INITIAL AND MAIN INPUT ALLOCATION IMPROPER. OUTPUT SPECIFY HIGH LABEL, LOW FILE. MINMEMORY xxxx EXCEEDS MAXMEMORY zzzz. PRINTER AND PUNCH ON SAME SYNCHRONIZER. DISCONTINUE RUN.
	<u>Explanation:</u> Compile, Generator, or Multifile Run. After the RUN Control Card and preceding operating option control cards (if any) are read, a check

Halt
Number Message, Explanation, and Action

is made for correctness of operating options. One or more of the first four messages above will be typed to indicate each discrepancy.

These errors may result from:

1. Improper operating option control cards preceding the RUN Control Card in this run.
2. Improper operating option control cards following the RUN Control Card in the Systems Run that created this systems tape (similar messages were produced at that time; the run should have been repeated with correct cards).
3. A combination of both of the above conditions.

Action: To accomplish this run, supply corrected operating option control cards and start the run from the beginning. If necessary, create a new systems tape, supplying correct operating option control cards.

0010 IMPROPER MULTIFILE ALLOCATION.
LIST TAPE IS WORK TAPE xx
LIST TAPE IS NOT PROVIDED
DECK TAPE IS WORK TAPE xx
DECK TAPE IS NOT PROVIDED
MAIN INPUT IS WORK TAPE xx
MAIN INPUT IS LIST TAPE xx
MAIN INPUT UNIT IS DECK TAPE xx
LIST TAPE IS DECK TAPE xx
(The above line will appear only if the MIXEDTAPE option NO has been specified.)
DISCONTINUE RUN.

Explanation: Multifile Run. When the RUN Control Card is read, a check is made for correctness of operating options. If any discrepancies are discovered, the systems tape is rewound and the header line above is typed. This line is followed by one or more of the subsequent lines listing each discrepancy.

These errors may result from:

1. Improper operating option control cards preceding the RUN Control Card in this run.
2. Improper operating option control cards following the RUN Control Card in the Systems Run that created this systems tape (similar messages were produced at that time; the run should have

Halt
Number Message, Explanation, and Action

been repeated with corrected control cards).

3. A combination of both of the above conditions.

Action: To accomplish this run, supply correct operating option control cards and start the run from the beginning. If necessary, create a new systems tape, supplying correct operating option control cards.

0011 MACRO TABLE IS FULL, NEW MACRO CANNOT BE ENTERED.
DISCONTINUE RUN.

Explanation: Generator Run. The new macro generator has been compiled, Halt 0067 has previously occurred, and the Start key has been pressed to enter the macro generator on a new systems tape. The related tables are full, however, so the new macro generator cannot be entered without first making a Systems Run to eliminate one of the existing macro generators.

Action: Discontinue the run, and save the current systems tape.

0067 LISTING ON TPxx.
PROGRAM ON TPxx.
COMPILATION OF MACRO COMPLETED.
TO ENTER IT IN LIBRARY, START.

Explanation: Generator Run. The compilation of the new macro generator is complete.

Action: To do an automatic Systems Run, entering this macro generator on the new systems tape, press START.

0070 RUN DISCONTINUED

Explanation: Any run. A message will have been typed preceding this halt. That message describes a discrepancy which makes it impossible to continue the current run. No usable output has been produced.

Action: This run cannot be accomplished until the indicated discrepancies are corrected.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>	<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0076	<p>LISTING ON TPxx. PROGRAM ON TPxx. NEW SYSTEMS TAPE ON TPxx. RUN COMPLETED.</p> <p><u>Explanation:</u> Any run. This halt signifies the completion of the current run (in the case of a Multifile Run, all compilations are complete). The last message is always typed. The other messages will occur where appropriate to the particular run type.</p> <p><u>Action:</u> Remove the specified output reels; label and save them. In the case of a Systems Run or Generator Run, save the old systems tape, at least until the correctness of the new tape is established.</p>		<p>deletion of a complete file as a result of invalid input.</p> <p><u>Action:</u> Discontinue the run.</p>
0800	<p>WORKING MEMORY OVERFLOW. DISCONTINUE RUN.</p> <p><u>Explanation:</u> COBOL. This halt indicates an unusual working memory overflow resulting from a program error or machine failure.</p> <p><u>Action:</u> Discontinue the run.</p>	0905	<p>TPxx TOO SHORT. DISCONTINUE RUN.</p> <p><u>Explanation:</u> Any run, input/output. A reflective spot has been sensed while writing intermediate output tape xx.</p> <p><u>Action:</u> Replace the tape with a longer one, and start the run from the beginning.</p>
0900	<p>EXISTING TAPE FILE CANNOT BE LOCATED. DISCONTINUE RUN.</p> <p><u>Explanation:</u> COBOL. A necessary input file no longer exists, due to either a machine failure or the deletion of a file as a result of invalid input.</p> <p><u>Action:</u> Discontinue the run.</p>	0909	<p>TPxx TP WD ERR</p> <p><u>Explanation:</u> Any run, input/output. A tape word error on input tape xx persists after nine attempts at rereading.</p> <p><u>Action:</u> To reread nine more times, press START.</p>
0901	<p>WORKING MEMORY OVERFLOW. DISCONTINUE RUN.</p> <p><u>Explanation:</u> Commercial Translator. This halt indicates an unusual working memory overflow resulting from a program error or machine failure.</p> <p><u>Action:</u> Discontinue the run.</p>	0910	<p>SYSTEMS TAPE ERROR y</p> <p><u>Explanation:</u> Any run, input/output. An error or unusual condition has occurred in reading the Compiler Systems Tape, mounted on tape unit 10. The digit y is the condition code digit of the Final Status Word.</p> <p><u>Action:</u> To repeat the read operation, press the Start key. (If y is a "1", indicating a read error, nine attempts at rereading have been made. Nine more attempts will be made if the Start key is pressed.) If the systems tape cannot be read successfully, discontinue the run.</p>
0902	<p>EXISTING TAPE FILE CANNOT BE LOCATED. DISCONTINUE RUN.</p> <p><u>Explanation:</u> Commercial Translator. A necessary input file no longer exists, due to either a machine failure or the</p>	0911	<p>TPxx READ FAIL</p> <p><u>Explanation:</u> Any run, input/output. An error on tape xx persists after nine attempts at rereading.</p> <p><u>Action:</u> To make nine more attempts at rereading, press START.</p>
		0912	<p>TPxx WRITE FAIL</p> <p><u>Explanation:</u> Any run, input/output. An error on output tape xx persists after five attempts at skipping and writing.</p> <p><u>Action:</u> If the Start key is pressed, five more attempts to rewrite will be made.</p>

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0922	<p>LAB ERR-xx CDy</p> <p><u>Explanation:</u> Any run, input/output. An error or unusual condition persists after seven attempts at rereading or rewriting a label on tape xx. The digit y is the condition code digit from the Final Status Word for the tape. This halt may indicate that an input label was not originally written at the density specified to the Compiler Systems Tape.</p> <p><u>Action:</u> Discontinue the run.</p>
0933	<p>TPxx LLR TPxx SCLR TPxx EOS</p> <p><u>Explanation:</u> Any run, input/output. Long Length Record, Short Character Length Record, or End-of-Segment Condition has occurred on tape xx.</p> <p><u>Action:</u> Discontinue the run except in the case of the End-of-Segment Condition, which may be ignored by pressing the Start key. The latter action is not recommended.</p>
0955	<p>OUTPUT TAPE ON DRIVE xx IS FULL. TO CONTINUE, MOUNT NEW TAPE AND PRESS START.</p> <p><u>Explanation:</u> Any run, input/output. The reflective spot on the indicated listing or program deck output tape has been sensed.</p> <p><u>Action:</u> Mount new tape, and press START.</p>
0969	<p>CARD READ ERROR</p> <p><u>Explanation:</u> Any run, input/output. An error has occurred when reading a card. (If a 7500 Card Reader is used, the card involved is offset and is the last card in the stacker.)</p> <p><u>Action:</u> Reproduce the card, making corrections as necessary. Clear the card reader and reload the remaining cards beginning with the corrected card. Ready the card reader, and continue the program by pressing the console Start key.</p>

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
0988	<p>TPxx 1EOF yyyyy TRL COUNT zzzzz</p> <p><u>Explanation:</u> Any run, input/output. The number (yyyyy) of tape blocks from input tape xx does not agree with the block count (zzzzz) contained in the trailer label for the file. (Note: Error statistics may appear to the right of the first line of the message.)</p> <p><u>Action:</u> To ignore the discrepancy, press START.</p>
0998	<p>TPxx PURGE DATE +00000yyddd TODAY +00000yyddd</p> <p><u>Explanation:</u> Any run. The purge date (creation date plus retention cycle) of the tape mounted on work tape unit xx exceeds today's date as contained in location 0109.</p> <p><u>Action:</u> To use this reel despite the discrepancy, press the Start key. Otherwise, mount another reel on the unit, and press the Program Reset key and then the Start key.</p>
1000	<p>SOURCE-COMPUTER INCORRECT.</p> <p><u>Explanation:</u> COBOL. A source computer other than 7070 or 7074 has been specified.</p> <p><u>Action:</u> Press START to continue if 7070 or 7074 should have been specified.</p>
1100	<p>OBJECT-COMPUTER INCORRECT.</p> <p><u>Explanation:</u> COBOL. An object computer other than 7070 or 7074 has been specified.</p> <p><u>Action:</u> Press START to continue if 7070 or 7074 should have been specified.</p>
1100	<p>DIVISION HEADER OMITTED OR INVALID. DISCONTINUE RUN.</p> <p><u>Explanation:</u> Commercial Translator. This message may indicate that the first card read by the processor is not a valid division header card and, therefore, subsequent source program entries cannot be identified. For 5K Commercial Translator, it may also indicate that the divisions are not in the prescribed order.</p>

<u>Halt</u> <u>Number</u>	<u>Message, Explanation, and Action</u>	<u>Halt</u> <u>Number</u>	<u>Message, Explanation, and Action</u>
	<u>Action:</u> Add or correct the division header card, or arrange the divisions in the proper order.	1300	DIVISION HEADER OMITTED OR INVALID. <u>Explanation:</u> COBOL. The first card read by the processor is not a valid division header card. Therefore, subsequent entries in the source program cannot be identified. <u>Action:</u> If the source program is in card form, add or correct the division header card, reload and ready the card reader, and press START to continue. If the source program is on tape, discontinue the run.
1101	NO PROCEDURE OR DATA. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. An initial scan of the source program has failed to reveal any procedure statements or data (explicit or implicit). <u>Action:</u> Check the source program for the following errors. 1. Incomplete source program. 2. Missing or mispunched division header cards.		
1102	SCAN ERRORS. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. A program error or machine failure has resulted in an invalid scan. <u>Action:</u> Discontinue the run.	1400	NO PROCEDURE OR DATA. DISCONTINUE RUN. <u>Explanation:</u> COBOL. An initial scan of the source program reveals that no procedure statements or data (explicit or implicit) are present. This halt may be caused by an incomplete source program or missing or mispunched division header cards. <u>Action:</u> Add or correct the division header cards, or complete the source program deck.
1111	LESS THAN 35 SUBSEQUENT ENTRIES IN TAPE FILE DEFINITION. <u>Explanation:</u> Compile Run. Phase 1, Autocoder. The programmer has omitted one or more of the subsequent entries which must follow a tape file definition entry. <u>Action:</u> Discontinue the run.	2000	WORKING MEMORY OVERFLOW. DISCONTINUE RUN. <u>Explanation:</u> COBOL. A working memory overflow has occurred during the construction of lists containing (a) procedure-names referenced by PERFORM commands and (b) input/output information specified in the Procedure Division. <u>Action:</u> Reduce the number of PERFORM statements and/or files, which are defined in the source program.
1200	CARD READ ERROR. <u>Explanation:</u> COBOL. This halt indicates a machine error or a mispunched card. <u>Action:</u> In the case of a mispunched card make the necessary correction. Reload and ready the card reader, and press START to continue.		
1200	WORKING MEMORY OVERFLOW. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. The processing of File Specifications caused a working memory overflow. <u>Action:</u> Reduce the combined length of the File Specifications.	2100	WORKING MEMORY OVERFLOW. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. The processing of BEGIN SECTION statements, function names, and procedure names referred to by DO statements has caused a working memory overflow.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
	<u>Action:</u> Reduce the combined length of BEGIN SECTION statements and the number of functions and closed subroutines.
3000	REPEATED MEMORY OVERFLOW PREVENTS FURTHER PROCESSING. DISCONTINUE RUN. <u>Explanation:</u> COBOL. This halt occurs after ten procedure statements have been deleted because of a working memory overflow. <u>Action:</u> Reduce the length of the procedure statements.
3100	REPEATED MEMORY OVERFLOW PREVENTS FURTHER PROCESSING. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator, Phase III. This halt occurs after ten procedure statements have been deleted because of a working memory overflow. <u>Action:</u> Reduce the length of the procedure statements.
3344	CARD READ ERROR. <u>Explanation:</u> Commercial Translator, Phase IV. This halt indicates a machine error or mispunched card. <u>Action:</u> In the case of a mispunched card, make the necessary corrections. Reload and ready the card reader, and press START to continue.
4100	REPEATED MEMORY OVERFLOW PREVENTS FURTHER PROCESSING. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator, Phase IV. This halt occurs after ten procedure statements have been deleted because of a working memory overflow. It can be traced to one of two causes: (1) extremely long procedure statements; (2) an attempt to process procedure statements when working memory is already crowded, under the conditions described in halt 2100 above.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
	<u>Action:</u> If the cause of the halt is (1) above, shorten the procedure statements. If the cause of the halt is (2) above, reduce the length of BEGIN SECTION statements and the number of functions and closed subroutines.
4101	WORKING MEMORY OVERFLOW PREVENTS PROCESSING OF CLOSED SUBROUTINES. <u>Explanation:</u> Commercial Translator. The processing of BEGIN SECTION statements and functions has caused a working memory overflow. <u>Action:</u> Reduce the combined length of BEGIN SECTION statements and the number of functions and closed subroutines.
4102	UNUSUAL ERROR CONDITION. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator, Phase IV. In the process of recovering from a memory overflow, another memory overflow has occurred. <u>Action:</u> Discontinue the run.
5000	UNUSUAL ERROR CONDITION. DISCONTINUE RUN. <u>Explanation:</u> COBOL. In the process of recovering from a memory overflow, another memory overflow has occurred. <u>Action:</u> Discontinue the run.
5100	UNUSUAL ERROR CONDITION. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator, Phase V. In the process of recovering from a memory overflow, another memory overflow has occurred. <u>Action:</u> Discontinue run.
6000	STATEMENT VERB CANNOT BE IDENTIFIED. DISCONTINUE RUN. <u>Explanation:</u> COBOL. As a result of a program error or machine failure, the verb of a procedure statement cannot be identified. <u>Action:</u> Discontinue the run.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
6100	STATEMENT VERB CANNOT BE IDENTIFIED. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. As a result of a program error or machine failure, the verb of a procedure statement cannot be identified. <u>Action:</u> Discontinue the run.
8000	FINAL AUTOCODER FILE CANNOT BE CONSTRUCTED. DISCONTINUE RUN. <u>Explanation:</u> COBOL. This halt indicates a machine failure. <u>Action:</u> Discontinue the run.
8100	ERROR MESSAGES CANNOT BE PROCESSED. DISCONTINUE RUN. <u>Explanation:</u> COBOL. A program error or machine failure has prevented the processing of the error message file. <u>Action:</u> Discontinue the run.

<u>Halt Number</u>	<u>Message, Explanation, and Action</u>
8100	FINAL AUTOCODER FILE CANNOT BE CONSTRUCTED. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. This halt indicates a machine failure. <u>Action:</u> Discontinue the run.
8200	ERROR MESSAGES CANNOT BE PROCESSED. DISCONTINUE RUN. <u>Explanation:</u> Commercial Translator. A program error or machine failure has prevented the processing of the error message file. <u>Action:</u> Discontinue the run.

Note: In the event of an unexpected stop, or any halt which can result from a program error, the Applied Programming Analysis Report, form 120-0482-0, should be submitted. A copy of the source program deck should accompany the material requested on the report.

MESSAGES

The following messages may be typed during a run which uses the Compiler Systems Tape. They are listed in approximate alphabetic order. The text of each message is shown in capital letters; an explanation follows, with a description of any action that may be taken by the console operator.

When messages are typed indicating errors or inconsistencies, the operator has the option of discontinuing the run immediately or of completing the run. It may be advisable to complete the run in order to detect other errors, but careful control should be kept over the existence and contents of any output produced by this run.

Note: Messages that may be typed during the execution of an object program produced from a FORTRAN source program are listed on page 47.

xxTH PROGRAM COMPILED

Explanation: Multifile Run. Compilation of the xxTH program has been completed. The Compiler Systems Tape will now compile the next program (if any).

Action: Remove and identify any on-line output.

xxTH PROGRAM DISCONTINUED

Explanation: Multifile Run. A preceding message has described a discrepancy in the indicated program which makes it impossible to continue the current compilation. The next program (if any) will now be compiled.

BRANCH CONTROL +01xxxxxxxx IGNORED

Explanation: Systems Run. In a PATCH or ENTER operation, an execute card is followed by condensed load cards. The execute card is ignored.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

COMPILE RUN

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Explanation: The first line identifies a Compile Run. It is followed by the name of the language as it appears in the control card, i. e., AUTO-CODER, FORTRAN, or REPORT GENERATOR, COMMERCIAL TRANSLATOR, or COBOL.

CONTROL CARD xxxxxxxxxxxxxxxx INVALID, IGNORED

Explanation: Systems Run. The operation portion of a control card is other than ENTER, DROP, or PATCH. (OBJCT or EQUIP are, however, valid until the first of these is reached.) The Xs are replaced by the contents of columns 6 through 20 of the invalid card.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

CONTROL CARD xxxxxxxxxxxxxxxx OUT OF SEQUENCE, IGNORED

Explanation: Systems Run. The label portion of a DROP, ENTER or PATCH Control Card is not in sequence with that of the previous valid and in-sequence control card. The Xs are replaced by the contents of columns 6 through 20 of the invalid card.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

DROP xxxxxxxxxxxx

Explanation: Systems Run. A control card indicates that the coding block named (Xs above) is to be deleted during the creation of a new systems tape.

ENTER xxxxxxxxxxxx

Explanation: Systems Run. A control card indicates that the coding block named (Xs above) is to appear on the new systems tape.

ERRORS IN INPUT, TO CONTINUE HIT START TO COMPILE NEXT PROGRAM HIT PROGRAM RESET AND START

Explanation: Compile Run or Multifile Run, FORTRAN. This message appears at the end of section 7 of FORTRAN if one or more errors in input have been detected.

Action: Press START to continue or, to compile the next program in a Multifile Run, press PROGRAM RESET and then START.

EXCESSIVE SEGMENTATION. PREVIOUS OPERATION IGNORED.

Explanation: Systems Run. An ENTER or PATCH operation has caused the number of contiguously-loaded segments of a coding block to exceed the number of RDWs permissible, i.e., 13. If the operation is ENTER, the coding block will not appear on the new systems tape. If the operation is PATCH, the coding block on the current systems tape will be written, unchanged, on the new systems tape, i.e., all patches for this particular coding block will be ignored.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

EXECUTE CARD IGNORED

(The contents of the card will be typed on this line.)

Explanation: Systems Run. An execute card follows a PATCH or ENTER Control Card, but an operation code other than Branch (+ 01) appears in columns 1 and 2. The card is ignored.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

GENERATOR RUN xxxxx

Explanation: This message identifies a run in which a macro generator (identified by Xs above) is to be compiled.

ILLEGAL PUNCH IN COLUMN 65. CARD IGNORED.

(The contents of the card will be typed on this line.)

Explanation: Systems Run. A condensed load card, following a PATCH or ENTER Control Card, contains a value greater than six in column 65. The condensed load card is ignored.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

MULTIFILE RUN

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Explanation: This message identifies a Multifile Run. It will be followed by the name of the source language prior to the compilation of each program.

NEW SYSTEMS TAPE IS VERSION xxxxx, CHANGE LEVEL yyyy.

Explanation: IBM-originated Systems Run. This message displays the version and change level of the new Compiler Systems Tape.

NO LOAD CARDS PRESENT. PREVIOUS OPERATION IGNORED.

Explanation: Systems Run. An ENTER Control Card is not followed by condensed load cards. No operation is performed on the new systems tape.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

NO SUCH NAME ON OLD SYSTEMS TAPE. PREVIOUS OPERATION IGNORED.

Explanation: Systems Run. A DROP or PATCH operation references a coding block which does not appear on the current systems tape. No operation is performed on the new systems tape.

Action: Check for correct punching of the name of the coding block. If the name is punched correctly, this message may indicate faulty control over the contents of the systems tape. If the name is punched incorrectly, this message usually indicates that another Systems Run, with corrected input data, will be required.

NOTE INITIAL INPUT UNIT A IS WORK TAPE xx.

NOTE INITIAL INPUT UNIT B IS WORK TAPE xx.

NOTE MAIN INPUT UNIT IS WORK TAPE xx.

NOTE INITIAL AND MAIN INPUT ALLOCATION IMPROPER.

NOTE INPUTS SPECIFY HIGH LABEL, LOW FILE.

NOTE OUTPUTS SPECIFY HIGH LABEL, LOW FILE.

NOTE MINMEMORY xxxx EXCEEDS
MAXMEMORY zzzz.

NOTE PRINTER AND PUNCH ON SAME
SYNCHRONIZER.

Explanation: Systems Run. When all of the permanent operating option control cards (if any) have been read, a check is made for correctness of the operating options to be included in the new systems tape. If the options are not correct, one or more of the messages above will be typed, detailing each error.

Action: Permitting the run to continue may be desirable in order to check any subsequent DROP, ENTER, or PATCH operations. However, the systems tape produced will include the errors noted. In the case of errors concerning initial input, close examination of the error conditions may indicate that the new systems tape cannot be used to initiate any type of run. The current systems tape must be saved so that another Systems Run can be made with corrected operating option control cards.

NOTE ON-LINE DECK ONLY

Explanation: Multifile Run. No program deck tape is specified; therefore, the program deck will be furnished on-line only.

NOTE xxxxxxxxxxxxxxxxxxxxxxxx INVALID, IGNORED

Explanation: Any run. The option card identified by the message is not valid and will be ignored. This message may apply either to cards which effect temporary options in any type of run or to cards which effect permanent options in a Systems Run; in the second case, the message "OPERATING OPTION CHANGES" will have been typed previously. The effect of ignoring this option card should be considered in deciding whether or not to continue the run.

Action: If the run is to be discontinued, press STOP.

OPERATING OPTION CHANGES

xxxxxxxxxxxEQUIP xxxxx

xxxxxxxxxxxOBJCT xxxxx

Explanation: Systems Run. If operating option control cards follow the RUN Control Card, the first line above is typed. One or more lines follow which duplicate the pertinent information

from each control card, thus furnishing a record of the changes.

PATCH xxxxxxxxxxxx

Explanation: Systems Run. A control card indicates that the coding block named (Xs above) is to be modified during the creation of a new systems tape.

REMOVE TPxx, MOUNT WORK TP. PRE-SORT

PRG NOW COMPILING. AT COMPLETION

RE-MOUNT TP12, RPT PRGM INPUT FOR A - 76.

Explanation: Report Program Generator Run. Two programs have been generated by the Report Program Generator to be used before and after a sort. The first (pre-sort) program is now being compiled into machine-language instructions. Tape unit xx contains the second (post-sort) program and a copy of the first (pre-sort) Report Program Generator symbolic program, in that order and separated by a tape mark. The tape has been written with label and density characteristics identical to those specified for the listing tape.

Action: Remove the tape; it must later be used as input for an Autocoder compilation of the second (post-sort) Report Program Generator program. Ready a work tape on tape unit xx.

REMOVE TP xx, MOUNT WORK TP. PRINT 12
FOR RPT PRGM COPY.

Explanation: Report Program Generator Run. The single program generated by the Report Program Generator is now being compiled into machine-language instructions. A copy of the symbolic generated program is on tape unit xx; this tape will be rewound and unloaded. The label and density characteristics of the tape containing the symbolic generated program are identical to those specified for the listing tape.

Action: If the tape is to be saved for listing or other use, remove it and mount a work tape on tape unit xx; if the tape is to be used as a work tape, reload it on the unit.

SORT HASH TOTAL OFF

SORT RECORD COUNT OFF

SORT SEQUENCE COUNT OFF

Explanation: Compile, Generator, or Multifile Run. One of the three messages above will be typed, indicating a corresponding error in one of several tape sorts used during compilation. The messages indicate discrepancies between the record count, hash total, or sequence count maintained in the current pass of the sort and that maintained in the previous pass. This condition is beyond the control of the Compiler Systems Tape, so the current compilation will be terminated. In a Multifile Run, the message "xxTH PROGRAM DISCONTINUED" will be typed, and the next source program (if any) will be compiled. In the other runs, the message "RUN DISCONTINUED" will be typed, followed by Halt 0070.

STORAGE CAPACITY EXCEEDED. PREVIOUS OPERATION IGNORED

Explanation: Systems Run. An ENTER or PATCH operation has caused the development of a coding block that exceeds the amount of storage available. If the operation is ENTER, the coding block will not appear on the new systems tape. If the operation is PATCH, the coding block on the current system's tape will be written, unchanged, on the new systems tape; i. e., all patches for this particular coding block will be ignored.

Action: This condition usually indicates that another Systems Run, with corrected input data, will be required.

SYSTEMS RUN

Explanation: This message identifies a Systems Run. Further messages will record the changes, if any, made during the creation of the new systems tape.

SYSTEMS RUN COPY

Explanation: This message identifies a Systems Run which will produce an exact duplicate of the current systems tape.

SYSTEMS RUN IBM

Explanation: This message identifies an IBM-originated Systems Run. Further messages will record the changes made during the creation of the new systems tape.

THIS AND ALL FURTHER MACRO CHANGES NOT REFLECTED IN TABLE

Explanation: Non-IBM Systems Run. More than one macro generator is being dropped. The name of each is typed. After the second name is typed, this message is issued to indicate that the names of all but the first macro generator dropped will remain in the macro table. If its DROP Control Card is the first Systems Update Control Card in the input deck, the name of the first macro generator will be dropped from the table; otherwise, this name too will remain in the table.

VERSION xxxxxx, CHANGE LEVEL yyyyy

zzzzzzzzzz

zzzzzzzzzz

etc.

Explanation: Any run. If alteration switch 3 is ON at the beginning of any run, the first line above is typed. In addition, during a Systems Run only, this line will be followed by the full ten-character name of each coding block written on the new systems tape.

APPENDIX A: CHECKPOINT IN COBOL AND
COMMERCIAL TRANSLATOR

When the Environment Division of a COBOL or Commercial Translator source program indicates that checkpoints are to be taken, one of the things generated by the processor is a linkage to the load program. This linkage supplies IOCS with the address of the first executable instruction of the load program. This address is 0308 for the standard load program stored in the standard storage locations (0300-0323).

If the load program used at any installation is either (1) not the standard load program or (2) the standard load program relocated in core storage, it is necessary to modify the systems tape in order to provide the processor with the appropriate load program address.

This modification is accomplished by means of a patch (see page 10) to the appropriate coding block shown in the following list.

<u>System</u>	<u>Coding Block Name</u>	<u>Address</u>
COBOL	C11Ab1Ab1S	1478
Commercial Translator (5K)	577Cb7Cb7S	2288
Commercial Translator (10K)	X75bb7P2CS	3519

The PATCH Control Card must be followed by a card containing the instruction:

+000000XXXX

where XXXX is the address of the first executable instruction of the user's load program.

APPENDIX B: LOADING THE FORTRAN OBJECT PROGRAM

The output from the FORTRAN processor (i. e., the object program) must be in relocatable form before it can be loaded.

The loading of the object program, either full or Basic FORTRAN, into core storage is accomplished by the FORTRAN Relocatable Loader. This is a separate program (not to be confused with the standard Condensed Card Load Program used to load the compiler) designed specifically to load the object program, the FORTRAN Package, and the user's subprograms and subroutines, if any are required. The Loader consists of the following cards:

1. The first card is a control card which may be replaced by the operator.
2. The remaining cards contain the routines which load the object program.

The FORTRAN Relocatable Loader must be preceded by the standard Condensed Card Load Program, which loads it into core storage. Once the Loader has been stored in the machine, it will load the FORTRAN package, the object program, and the user's subroutines and/or subprograms.

INPUT TO THE LOADER

The Relocatable Loader and the object program deck may be in the form of cards or in the form of card images on tape. Input can be mixed, e. g., the Loader and several library functions may be on tape, while the main program is on cards. Up to six different input units may be used to load the package and the object program. The second through the sixth input units are specified by the control word IN (see "Control Words," page 45).

If the object program is to be run again and again, it may be somewhat inefficient to load it each time with the Relocatable Loader. Therefore, the operator can elect to have the Relocatable Loader produce a relocated deck (or tape) preceded by (1) the standard Condensed Card Load Program, which is sufficient to load it, and (2) a clear-storage card. This feature also makes it possible to load object programs that fit into core storage only when the Relocatable Loader is not in storage.

THE RELOCATABLE LOADER

A relocatable loader is a program which stores a series of relocatable routines in core storage. In order to do this, the loader performs the following functions:

1. Records the number of routines loaded and the number of locations each routine requires.

2. Notes the address of the first location (the base address) of each routine as it was originally assembled.

3. Computes new addresses for each routine in order to place the routine in any given part of storage as needed.

4. Constructs linkages where needed so that the routines can work together.

At object time, all assembled FORTRAN pro- and other required routines (subroutines or subprograms) are loaded into storage by the FORTRAN Relocatable Loader. So that the Loader can handle each routine, certain conventions must be followed in routines not compiled from 7070 FORTRAN.

REQUIREMENTS OF RELOCATABLE ROUTINES

Each routine handled by the Loader must meet the following requirements:

1. All routines loaded must be preceded by a Title Card.
2. If a routine calls or refers to any other routine (function or subprogram), the calling program cards must include, in the proper place, a Transfer Entry Card for each called routine.
3. Each routine loaded must have an Execute Card as its last card.
4. All routines loaded by the Relocatable Loader must be punched in the standard Autocoder Relocatable Condensed Card format.
5. Each routine that calls another routine must have space reserved in it for a Branch List which is constructed by the Loader with Transfer Entry Cards. All necessary Title Cards, Transfer Entry Cards, Branch Lists, etc., are automatically supplied to any program compiled from the FORTRAN language.

The Title Card

The Title Card is compiled as the first card of every routine processed by FORTRAN. It must be supplied by the user for every routine loaded by the Relocatable Loader which has not been compiled from FORTRAN. The Title Card may be supplied by the user in actual form at object time, if it is not supplied in symbolic form at compile time. The actual Title Card is described below.

Card Columns

1-20

21-30

Contents

The name of the routine, left-justified, in double-digit form. (See Example below.)

The number of locations to be reserved for the data area

	following the last instruction of the routine (right-justified).
31-40	The number of locations to be reserved at the top of storage for the common data area (right-justified).
41-50	The relocation base(right-justified).
51-60	Zeros.
61-62	This must contain 91 which identifies this card to the Loader as a Title Card.
63-69	Zeros.
70	The digit 0, 1, or 2, as explained under "Relocation Indicators," in the FORTRAN Reference Manual, form C28-6170.
71-75	Card Number.
76-80	Reserved for program identification.
10, 20, 30, 40, 50, 60, 70, 79, 80	12-punches.

Example: A subroutine called SINF would be entered into columns 1-20 in this form:

```

82 69 75 66 00 00 00 00 00
S I N F b b b b b b

```

Transfer Entry Cards

Transfer Entry Cards are used by the Relocatable Loader to complete linkages between routines after they have been loaded into storage. If a subroutine uses or calls no other subroutines, no Transfer Entry Cards are included. The Transfer Entry Card may be supplied by the user in actual form at object time if it is not supplied in symbolic form at compile time. The actual Transfer Entry Card is described below.

<u>Card Columns</u>	<u>Contents</u>
1-20	Name of the called routine, left-justified, in double-digit form.
21-60	Zeros.
61-62	This must contain 92, which identifies this card to the Loader as a Transfer Entry Card.
63-70	Zeros.
71-75	Card Number.
76-80	Reserved for program identification.
10, 20, 30, 40, 50, 60, 70, 79, 80	12-punches.

Execute Cards

Every routine loaded by the Relocatable Loader must have an Execute Card as its last card. The format of the Execute Card is as follows:

<u>Card Columns</u>	<u>Contents</u>
1-10	A branch to the first instruction of the routine.
11-55	Zeros.
56	The relocation indicator for the branch instruction in columns 1-10.
57-60	Zeros.
61-62	Zeros, which indicate to the Loader that this is an Execute Card.
63-70	Zeros.
71-75	Card Number.
76-80	Reserved for program identification.
10, 20, 30, 40, 50, 60, 70, 79, 80	12-punches

Special Execute Card

This special execute card is provided for the convenience of users who wish to place data for the object program directly after a program on an input unit. The special execute card may be placed at the end of a program and followed immediately by object program data, which is to be read from the same input unit. The Loader regards the special execute card as a substitute for a physical end-of-file condition. The use of the special execute card allows object program data to be placed into a card reader at the same time as a program deck; i. e., there is no need to let the card reader hopper run out of cards to provide an end-of-file condition after the program deck has been loaded. Similarly, when placing a program and object program data onto tape by a card-to-tape operation, use of the special execute card eliminates the need of writing a tape mark between the program and the data for the object program.

The special execute card is to be punched as follows:

<u>Card Columns</u>	<u>Contents</u>
1-60	Zeros.
61-62	93.
63-80	Zeros.
10, 20, 30, 40, 50, 60, 70, 79, 80	12-punches.

Note: When a relocatable program deck or tape (produced by the Relocatable Loader) is being loaded, a special execute card must not be used; the relocated program is loaded by the standard Condensed Card Load Program rather than by the Relocatable Loader, which detects the special execute cards. Data for the object program should be placed immediately after the relocated program if it is to be read from the same unit; the execute card produced at the end of the relocated program will cause the object program to be executed without interruption.

THE LOADING PROCEDURE

The Relocatable Loader loads itself into the highest available locations in storage. It then places the FORTRAN Package into lower storage, and, finally, loads the object program into storage. An end-of-load condition may be caused by one of the following:

1. An end-of-file condition on an input unit, and zeros in the digit positions subsequently examined in control word IN.
2. An end-of-file condition on the sixth input unit to be used by the Loader.

ASSUMPTIONS MADE BY THE RELOCATABLE LOADER

The Relocatable Loader and the FORTRAN Package assume certain conditions to be true at object time. If these conditions are not true, certain control information, explained later, must be provided by the user. The assumptions are as follows:

1. The package and the object program will be read from the same input device that read the Loader.
2. If the Loader is producing the relocated program (see page 43), the output will be punched.
3. The highest location in storage available to the object program is 9989.
4. The object program will begin immediately after the FORTRAN Package.
5. A 7500 Card Reader is on Synchronizer 1.
6. A 7550 Card Punch is on Synchronizer 1.
7. A 7400 Printer is on Synchronizer 2.

CONTROL WORDS

This section explains the uses of several words of storage in the Loader and Package which control certain operating options. If options other than those assumed are desired, the word or words must be patched.

Control Words in the Relocatable Loader

IN. If the package or any part of the object program is to be read from an input unit other than the one which read the Loader, or from another file on that same unit, the operator must patch the control word IN.

When the Loader reaches an end-of-file condition on an input unit, it checks IN to see if there is more input on another unit. At the first end of file, it interrogates digit positions 0 and 1. If they are not zero, the Loader continues reading from the indicated unit. If the digits are zero, the Loader assumes an end-of-load condition. If more input is indicated, then the next time an end-of-file condition is reached, digit positions 2 and 3 of IN are examined, and so on. Thus, input to the Loader may come from up to six different input units. All tapes must have tape marks.

The format of the control word IN follows:

<u>Digit Positions</u>	<u>Contents</u>
0-1	The synchronizer (0s) or the tape channel and unit (cu) from which additional input is to be read after the first end-of-file condition is sensed.
2-3	The source of input after the second end-of-file condition is sensed.
4-5	The source of input after the third end-of-file condition is sensed.
6-7	The source of input after the fourth end-of-file condition is sensed.
8-9	The source of input after the fifth end-of-file condition is sensed. By adding 50 to the number of the input unit, a library file can be indicated. The loader will extract from this file only the programs previously called for.

Example: If IN contained +1212022100, loading would proceed as follows:

1. The Relocatable Loader, the FORTRAN Package, and any routines following them, would be read from the input unit referred to in locations

0000, 0001, and 0002. It is not necessary for anything other than the loader to be read from that unit.

2. After an end of file was reached on that unit, loading would continue from tape unit 12.

3. After an end of file on tape unit 12, loading would continue with a second file from tape unit 12.

4. After a second end of file on tape unit 12, loading would continue from synchronizer 2.

5. After an end of file on synchronizer 2, those library routines previously called for would be loaded from tape unit 21. This would be taken as the last input source.

OUTLOWHIGH. The control word named **OUTLOWHIGH** has the format **+xyyyzzzz** and provides the following information to the Loader:

1. Digit positions 0 and 1 contain either the number of the synchronizer or the numbers of the tape channel and unit on which a relocated deck will be written in the numerical mode (provided that alteration switch 3 is ON). To obtain a tape written in the alpha mode, add 50 to the tape channel and unit number placed in these positions. Note that 50 is not added to a tape channel and unit number in the control word **IN** for that purpose; the reading of numerical or alpha tapes is done automatically.

2. Digit positions 2 through 5 (yyyy above) contain the lowest address to be occupied by the object program (not including the package).

Since the package is not relocatable and must not be overlaid in storage by the object program, positions 2-5 of word **OUTLOWHIGH** are treated as follows: If, after the Loader and Package have been placed in storage, these positions contain an address lower than the first address following the Package, the specified address will be ignored, and the object program will be placed in storage immediately following the Package. If positions 2-5 of **OUTLOWHIGH** contain an address greater than the last address occupied by the Package, the object program will be placed in storage beginning at the specified address. This feature makes it possible to leave room for patches to the Package in relocated decks written by the Loader.

3. Digit positions 6 through 9 (zzzz above) contain the highest address available to the object program, including data. (This will also be the highest address occupied by the Loader during the loading procedure and the highest zeroed by the clear-storage card in the relocated program written by the Loader.)

It should be noted that this address will not cause a relocation downward for programs compiled by FORTRAN prior to October 1961.

Preparation of the Loader Control Card

If it is necessary to change the assumptions specified by **IN** or **OUTLOWHIGH**, the first two words of the first card of the Loader deck must be changed; the remainder of the card must be duplicated without change. This control card should be added to the Loader as follows:

Remove the first card of the Loader deck and replace it with the new control card.

The contents of the Loader Control Card are as follows:

<u>Card Columns</u>	<u>Contents</u>
1-10	The information to be stored in control word IN .
11-20	The information to be stored in control word OUTLOWHIGH .
10-20	12-punches.
21-80	Duplicated from first card of the Loader deck.

Control Words in the FORTRAN Package

UNITSYN COP. The word **UNITSYN COP** is composed of

+r1p2000000

The read synchronizer used by the object program is represented by **r**, and the punch synchronizer used by the object program is represented by **p**.

PRINT1 and **PRINT2.** The printer synchronizer is controlled by two words. The synchronizer specified by both words must be the same.

The word **PRINT1** is composed of

+6900w2xxxx

The word **PRINT2** is composed of

+6900w2yyyy

In each of these words, **w** is the printer synchronizer.

The address of the **RDW** being used is specified by **xxxx** in **PRINT1** and by **yyyy** in **PRINT2**.

Patching the Package Control Words

In the above control words, **r** and **p** are assumed to be 1 and **w** is assumed to be 2. If these assumptions are to be changed, the appropriate word must be patched by means of a standard condensed card. The absolute locations and contents of these words

are listed in the write up furnished with the package.

LOADING SETUP AND OPERATION

Control Panels Required

The control panels required during the loading and running of the object program are the 7070 utility control panels (same as those used during compilation). Wiring diagrams for these panels are contained in the IBM Data Processing System Bulletin "IBM 7070 Utility Control Panels," form J28-6095.

7500 Card Reader

If any part of the object program is to be loaded from cards, set alteration switches as follows:

1. Alteration switch 1:A
2. Alteration switch 2:B.
3. The positions of switches 3 and 4 do not matter.

Press the Start key and the End-of-File key.

7400 Printer

Set all four alteration switches to A and depress the Start key.

7550 Card Punch

Set the alteration switches as follows:

1. Alteration switch 1: this setting depends on setting of console alteration switch 3: (see below).
2. Alteration switch 2:B.
3. The positions of switches 3 and 4 do not matter.

Place blank cards in the punch hopper and press START.

729 Tape Units

Mount and ready any tapes used to load the object program or used by the object program.

7150 Console

Set the console operating keys and dials as follows:

1. Alteration switches 1 and 2 are not used.
2. Alteration switch 3 should be set ON if a relocated deck or tape is desired. Otherwise, alteration switch 3 should be set OFF. (See page 7.)

Note: If alteration switch 3 is ON, and the relocated deck is to be punched in cards, alteration

switches 1 and 2 on the card punch should be set to AB during the loading procedure and then changed to BB at end of load if the punch is to be used by the object program. If alteration switch 3 is OFF, alteration switches 1 and 2 on the card punch should be set to BB.

3. If alteration switch 4 is set ON, the following information regarding the allocation of storage will be typed: the name, the starting location, and the first data location (if any) of each routine loaded. Upon completion of loading, the address of the beginning of the COMMON data area (if any) will be typed.

4. Accumulator and Exponent Overflow keys: ON.

5. Priority Channel A and Priority Channel B dials: OFF and N.

Initiating the Loading Process

To read the Relocatable Loader into storage, proceed as follows:

1. a. If the loader is to be read from cards, manually store the following instructions:

<u>Location</u>	<u>Contents</u>
0000	+6901s10004
0001	-0100030010
0002	+0000000002

The number of the synchronizer for the card reader that contains the Relocatable Loader is represented by s above.

- b. If the Loader is to be read from tape, manually store the following instructions:

<u>Location</u>	<u>Contents</u>
0000	-8c01u10004
0001	-0100030010
0002	+5100c10002

The channel number of the Loader tape is represented above by c.

The unit number of the Relocatable Loader tape is represented by above u.

2. Press the Run Status key.
3. Press the Computer Reset key.
4. Press the Start key.

FORTTRAN Loader Halts and Messages

The following list contains the messages which may be typed by the Relocatable Loader. Most of the

messages will be accompanied by a procedural halt to allow the operator to take the necessary action.

xxxxx DUPLICATE BYPASSED

Explanation: Two routines, each with the name xxxxx, have been encountered by the Loader. The first one has been loaded, and the second one has been bypassed.

MAINPROG DUPLICATE BYPASSED

Explanation: The Loader has found two routines named MAINPROG. It has loaded the first and will bypass the second.

Note: Each object program must have one and only one routine named MAINPROG.

Action: Press START to continue loading, thereby discarding the second MAINPROG.

MAINPROG MISSING

Explanation: The Loader has completed the loading process and has not found a routine named MAINPROG.

Note: Each object program must contain one and only one routine named MAINPROG.

xxxxx MISSING
UNEXECUTABLE PROGRAM

Explanation: The Loader has completed the loading process and has not encountered a routine named xxxxx which is necessary to the execution of the program. If more than one routine is missing, a message will be typed for each.

Action: Ready the missing routines at the appropriate input unit, and press the Start key on the console.

xxxxx NOT USED

Explanation: A routine has been loaded which evidently is not needed for the execution of the object program. If this is the case, the program is executable. However, this message could be caused by a missing Transfer Entry Card.

Action: If an extra routine has been loaded, it will not affect the execution of the object program in any way, and the program may be started. If, however, a Transfer Entry Card is missing, the program is unexecutable. The missing card must be replaced properly, and the routine containing it must be reloaded.

PROGRAM LOADED

Explanation: The Relocatable Loader has successfully loaded the object program.

Action: If necessary, reset console alteration switches for the requirements of the object program. If the card punch is to be used by the object program, reset alteration switches 1 and 2 to BB. Place data cards, if any, in the card reader. Check to see that all input/output units to be used by the object program are ready. Press START.

PROGRAM TOO BIG

Explanation: The object program has exceeded the storage capacity of the computer.

Action: The run must be discontinued.

PUNCH ERROR

Explanation: A punch error has occurred during the punching of a relocated object deck. The erroneous card has been offset in the stacker.

READ ERROR

Explanation: A card read error has occurred.

Action: The operator should correct the card in error, replace it in the reader, and press START to continue reading.

RELOAD WITH ALT 3 ON

Explanation: The object program would have overlaid the Relocatable Loader in upper storage if the entire program had been loaded directly; however, the object program is not too big for storage.

Action: The object program must be reloaded with alteration switch 3 ON in order to obtain a relocated deck. The relocated deck is loaded with the Condensed Card Load Program already on the front of the deck.

RELOCATED PROGRAM IS ON TAPE
TO RUN PRESS START

Explanation: The object program has not been completely loaded into storage because it would have overlaid the Relocatable Loader. However, because alteration switch 3 is ON, the complete relocated program is on tape.

Action: Press START. The object program will be completely loaded into storage, and execution will begin.

RELOCATED PROGRAM IS IN CARDS
TO RUN PRESS START

Explanation: The object program has not been completely loaded into storage because it would have overlaid the Relocatable Loader. However, because alteration switch 3 is ON, the complete relocated program is in card form.

Action: Press START. The object program will be completely loaded into storage, and execution will begin.

TITLE CD MISSING

Explanation: An attempt has been made to load a program that is not preceded by a Title Card.

Action: If input is in card form, supply the Title Card and press START to restart at the point at which this halt occurred. If input is in tape form, a new tape must be created with the missing Title Card included before this program can be loaded.

WRITE ERROR

Explanation: Ten attempts have been made to write out the relocated program.

Action: Mount new tape and start over.

Error Halts and Messages in the Object Program

The following messages may be typed during the execution of an object program produced from a FORTRAN source program.

TAPE ERROR d

Explanation: A tape error has occurred. The d above represents a numerical code from 0-7 corresponding to a condition code explained in the Reference Manual "IBM 7070 Data Processing System!" under "Tape Priority, Condition Codes."

Action: If d is either 0 or 1, pressing START will cause an attempt to read or write again. If d is not 0 or 1, the program cannot be restarted.

UNIT RCD EOF

Explanation: An end-of-file condition has occurred at either a card reader or a card punch.

Action: Pressing START will cause the program to read (or write) again.

UNIT RCD ERROR

Explanation: An error has occurred at a card reader, card punch, printer, or typewriter.

Action: The program cannot be restarted.

APPENDIX C: REPORT PROGRAM OPERATING PROCEDURES

The steps required to produce a report will depend on whether a pre-sort program has been generated in addition to the report program. Two sets of operating instructions are given below: the first is used when sorting is not required; the second is used when a pre-sort program, Sort 90, and then the report program are to be run.

Sorting Not Required

1. Mount the first reel of the input file on the main input tape unit; mount the second reel (if any) of this file on the alternate tape unit if one is available. Place the tape unit(s) in ready status. The tape channel and tape unit numbers may be obtained from columns 12 through 15 of the Tape File and Printer Specifications form for the report program to be executed. They may also be obtained from the symbolic listing of that program.

2. Mount a reel of tape on the main output tape unit; mount another reel on the alternate tape unit if one is available. Place the tape unit(s) in ready status. The tape channel and tape unit numbers may be obtained from columns 56 through 58 of the Tape File and Printer Specifications form for the report program to be executed. They may also be obtained from the symbolic listing of that program.

3. Load the report program. (If an on-line printer is to be used, see Note below.)

Sorting Required

1. Mount the input tape(s) as explained in step 1 under "Sorting Not Required."

2. Mount tapes to be used for the intermediate file on the tape units specified in columns 32 through 45 of the Tape File and Printer Specifications form for the report program to be executed. Place the tape units in ready status.

3. Load the pre-sort program.

4. After the message "END PRE-SORT: START SORT 90" has been typed, load Sort 90. The pre-sort program will leave the control data for sorting in storage; therefore, Sort 90 Modification I-1 should be used. (See "Phase I Modifications" in the Reference Manual "7070/7074 Generalized Sorting Program: Sort 90," form C28-6111-1.)

5. After sorting has been completed, mount the first reel of the sorted output on the main intermediate tape unit; mount the second reel, if any, of the sorted output on the alternate tape unit if one is available. Place the tape unit(s) in ready status. The tape channel and tape unit numbers may be obtained from columns 32 through 35 of the Tape File and Printer Specifications form for the report program to be executed.

6. Mount the output tape(s) as explained in step 2 under "Sorting Not Required."

7. Load the report program. (If an on-line printer is to be used, see Note below.)

Note: Whenever on-line printing is required, insert the 7400 Utility Panel, set the printer alteration switches to AAAA, position the paper in the carriage, and place the printer in ready status before loading the report program.

APPENDIX D: COMPOSITION OF THE COMPILER SYSTEMS TAPE

At the beginning of the Compiler Systems Tape, there are several condensed-card records that constitute a "bootstrap" loading procedure; at the end of the tape is a tape mark. In between are a series of coding blocks, each prefaced by a sixteen-word identification record. The coding blocks consist of the instructions and/or data to be loaded. All records are written in high density.

The format of an identification record is:

<u>Words</u>	<u>Contents</u>
1-2	Complete name of the associated coding block, in double-digit alphameric representation.
3-15	A list of as many as 13 RDWs defining the storage location into which the coding block is to be loaded.
16	A Branch instruction to be used, where necessary, for branching into the coding block after it has been loaded.

The identification record is created automatically during a Systems Run by means of an ENTER operation. It is also updated automatically during a PATCH operation.

All actions performed with or on the systems tape (e.g., loading, updating, etc.) require that the names of the several coding blocks be in strictly ascending sequence from first to last. All comparisons of the names of coding blocks are made on the basis of the ten alphameric characters which make up words one and two of the identification record. In every run, alteration switch 4 is checked at the time a coding block is loaded; if this switch is ON, the five alphameric characters in word two will be typed. During a Systems Run only, turning alteration switch 3 ON will cause the full ten-character name of each coding block to be typed as it is written on the new systems tape.

Below is a list of coding blocks and the first two words of their associated identification records as they appear on the Compiler Systems Tape supplied by IBM. The list is divided into nine columns, each indicating a type of run as follows:

1. Systems Run (Systems)
2. Generator Run (Generator)
3. Autocoder Compile Run (Autocoder)
4. Autocoder Compile Run for Additional Storage (Autocoder AS)
5. Report Program Generator Compile Run (RPG)
6. FORTRAN Compile Run (FORTRAN)
7. Commercial Translator Compile Run on 10K Machine (CT 10K)
8. Commercial Translator Compile Run on 5K Machine (CT 5K)
9. COBOL Compile Run (COBOL)

Each column has an X on the line next to each coding block used during the indicated type of run. In order to make a systems tape to be used exclusively for one type of run, the coding blocks that do not have an X in the column for that run may be removed by preparing a DROP Control Card for each, and doing a Systems Run. If this procedure is followed, it is recommended that a SYSTEMS RUN IBM Control Card be used so that the name of the tape may be altered to reflect the change. The new name should incorporate the original version and modification level numbers, i.e., a FORTRAN only tape made from Version 00003, Modification Level 00001, could be named Version 03F01, Modification Level 00000.

A systems tape modified in this fashion can be used only for the type of run that the coding blocks left on the tape are designed for. If another type of run is attempted, a SYSTEMS TAPE ERROR 5 will probably occur. This indicates that a call has been executed for a coding block that is no longer on the tape.

If it is anticipated that any type of Systems Run will be made using the modified systems tape, the coding blocks in the Systems Run category must be left on the new tape.

Description	Identification		Systems	Generator	Autocoder	Autocoder			CT 10K	CT 5K	COBOL
	Word 1	Word 2				AS	RPG	FORTRAN			
Systems Control 1	bbbb	SYCL1	X	X	X	X	X	X	X	X	X
Systems Control 2	bbbb	SYCL2	X	X	X	X	X	X	X	X	X
Communications Record	bbbb.	CMREC	X	X	X	X	X	X	X	X	X
	bbbb.	DEBUG	(Used in IBM runs only)								
Systems Control 3	bbbb.	SYCL3	X	X	X	X	X	X	X	X	X
Systems Control 4	bbbb.	SYCL4					X	X			
Systems Control 10	bbbb.	SYC10						X	X	X	X
Update	bbbb.	UPDAT	X	X							
Autocoder Phase I											
Generator Edit	A00bb	PREDT		X							
Macro Table	A10b*	PREMT		X							
Systems Control 5	A15bb	SYCL5		X							
Pass 1											
Coding block 1	A20bb	P1P1A		X	X	X	X		X	X	X
Coding block 2	A21bb	P1P1B		X	X	X	X		X	X	X
Macro Table	A30b*	P1MTA		X	X	X	X		X	X	X
Commercial Translator - 10K											
Phase 1A, Part 1											
System Control	C10bb	1A1CS							X		
Coding block 1	C10bb	1A1CT							X		
Coding block 2	C11bb	1A1CT							X		
COBOL Phase IA											
System Control	C11Ab	1Ab1S									X
Commercial Translator 5K											
System Control -											
Initial Scans	C71Ab	1Ab7S									X
FORTRAN											
Section 1	F10bb	FRT1A						X			
Section 1A	F13bb	FRT1B						X			
Report Generator											
Phase 1	G11bb	RPG11						X			
COBOL											
Phase 1A											
Coding block 1	K11A1	1A11T									X
Coding block 2	K11A2	1A21T									X
Coding block 3	K11A3	1A31T									X
Commercial Translator - 5K											
Phase 1A	K71Ab	1Ab7T									X
Phase 1B	K71Bb	1Bb7T									X
Phase 1C	K71Cb	1Cb7T									X
Autocoder Phase III											
Error Message	MbbM3	PH3MS		X	X	X	X	X	X	X	X

<u>Description</u>	<u>Identification</u>		<u>Systems</u>	<u>Generator</u>	<u>Autocoder</u>	<u>Autocoder</u>	<u>RPG</u>	<u>FORTRAN</u>	<u>CT</u>	<u>CT</u>	<u>COBOL</u>
	<u>Word 1</u>	<u>Word 2</u>				<u>AS</u>			<u>10K</u>	<u>5K</u>	
Autocoder Phase II											
Edit 2	MAA20	P2E2A		X	X	X	X	X	X	X	X
Sort 2	MAA41	PH2S2		X	X	X	X	X	X	X	X
Merge 2	MAA42	PH2M2		X	X	X	X	X	X	X	X
Control	MABbb	PH2CT		X	X	X	X	X	X	X	X
Macro Generators											
MACRO	A. DEF							X			
MACRO	A. GTO							X			
MACRO	A. RAY							X			
MACRO	A. SGN							X			
MACRO	ABSbb		(Used in IBM runs only)								
MACRO	AR. T2			X	X	X	X	X	X	X	
MACRO	ARITH			X	X	X	X	X	X	X	
MACRO	ASINb			X	X	X			X	X	X
MACRO	ATANb			X	X	X			X	X	X
MACRO	B. SPC							X			
MACRO	BSPbb				X	X			X	X	X
MACRO	C. GTO							X			
MACRO	CHANA				X	X	X		X	X	X
MACRO	CHPTb				X	X			X	X	X
MACRO	CHPTA				X				X	X	X
MACRO	CHPTB				X				X	X	X
MACRO	CHPTC				X				X	X	X
MACRO	CHPTD				X				X	X	X
MACRO	CLOSE				X	X	X		X	X	X
MACRO	CLSED				X						
MACRO	COMPb			X	X	X			X	X	X
MACRO	COSbb			X	X	X			X	X	X
MACRO	CPTE.				X				X	X	X
MACRO	CYCLE			X	X	X			X	X	X
MACRO	D. bbb							X			
MACRO	D. CSQ		(Used in IBM runs only)								
MACRO	DCHPT				X	X			X	X	X
MACRO	DECOD			X	X	X			X	X	X
MACRO	DEORb				X	X			X	X	X
MACRO	DIOCS				X	X	X		X	X	X
MACRO	DIV. b			X	X	X	X	X	X	X	X
MACRO	DRDWb			X	X	X			X	X	X
MACRO	DTEXT		(Used in IBM runs only)								
MACRO	DTFbb				X	X	X		X	X	X
MACRO	DTF1b				X	X	X		X	X	X
MACRO	DTF2b				X	X	X		X	X	X
MACRO	DUBL.			X	X	X	X		X	X	X

Description	Identification		Systems	Generator	Autocoder	Autocoder			CT 10K	CT 5K	COBOL		
	Word 1	Word 2				AS	RPG	FORTTRAN					
MACRO	DUFbb				X	X			X	X	X		
MACRO	DUMMY		(Used in IBM runs only)										
MACRO	E.DDO							X					
MACRO	E.DFL							X					
MACRO	E.DIO							X					
MACRO	EDITF			X	X	X			X	X	X		
MACRO	EDMOV			X	X	X	X		X	X	X		
MACRO	ENDbb				X	X			X	X	X		
MACRO	ENTDP				X								
MACRO	EORAb				X	X	X		X	X	X		
MACRO	EORBb				X	X	X		X	X	X		
MACRO	EORCb				X	X			X	X	X		
MACRO	EORDb				X	X			X	X	X		
MACRO	EXDPb				X								
MACRO	EXIGA					X							
MACRO	EXITb			X									
MACRO	EXOPA					X							
MACRO	EXOPB					X							
MACRO	EXOPC					X							
MACRO	EXOPD					X							
MACRO	EXPbb			X	X	X			X	X	X		
MACRO	EXP.b			X	X	X	X	X	X	X	X		
MACRO	EXPEb			X	X	X			X	X	X		
MACRO	F.TC1	}	(Used in IBM runs only)										
MACRO	F.TC2												
MACRO	F.TC3												
MACRO	F.TC4												
MACRO	FEORb				X	X					X		
MACRO	FEORN				X	X					X		
MACRO	FILLb			X	X	X			X	X	X		
MACRO	FIX.b			X	X	X		X					
MACRO	FLOT.			X	X	X		X					
MACRO	FLTF.			X	X	X			X	X	X		
MACRO	FLTSM			X	X	X		X	X	X	X		
MACRO	FORF.			X	X	X		X					
MACRO	G.TOb							X					
MACRO	GENER			X									
MACRO	GETbb				X	X	X		X	X	X		
MACRO	GETDb				X								
MACRO	H.Bbb							X					
MACRO	I.AOV							X					
MACRO	I.BRb							X					
MACRO	I.DVC							X					

<u>Description</u>	<u>Identification</u>		<u>Systems</u>	<u>Generator</u>	<u>Autocoder</u>	<u>Autocoder</u>			<u>FORTRAN</u>	<u>CT 10K</u>	<u>CT 5K</u>	<u>COBOL</u>
	<u>Word 1</u>	<u>Word 2</u>				<u>AS</u>	<u>RPG</u>					
MACRO	I. LGT								X			
MACRO	I. QOV								X			
MACRO	I. SWT								X			
MACRO	IGENA				X			X		X	X	X
MACRO	IGENB				X	X	X	X		X	X	X
MACRO	IGENC				X	X	X	X		X	X	X
MACRO	IGEND				X	X	X	X		X	X	X
MACRO	IGENE				X	X	X	X		X	X	X
MACRO	INTLK				X							
MACRO	L. BEL								X			
MACRO	L. ORG								X			
MACRO	L. STb								X			
MACRO	LINKb		X		X	X	X			X	X	X
MACRO	LMSR1				X	X				X	X	X
MACRO	LMSR2				X	X				X	X	X
MACRO	LOGbb		X		X	X				X	X	X
MACRO	LOGEb		X		X	X				X	X	X
MACRO	LOGIC		X		X	X	X			X	X	X
MACRO	MDWAb				X							
MACRO	MODL1			X								
MACRO	MODL2			X								
MACRO	MOVEb		X		X	X	X			X	X	X
MACRO	MOVES		X		X	X						
MACRO	NOP. b		(Used in IBM runs only)									
MACRO	NOTEb			X								
MACRO	OPENb				X	X	X			X	X	X
MACRO	OPENA				X		X			X	X	X
MACRO	OPENB				X		X			X	X	X
MACRO	OPENC				X		X			X	X	X
MACRO	OPEND				X							
MACRO	OPENE				X							X
MACRO	OPND.				X					X	X	X
MACRO	ORDER		X		X	X				X	X	X
MACRO	P. INT								X			
MACRO	P. NCH								X			
MACRO	P. USE								X			
MACRO	PUTbb				X	X	X			X	X	X
MACRO	PUTDb				X							
MACRO	PUTXb				X	X				X	X	X
MACRO	QOPEN		(Used in IBM runs only)									
MACRO	R. Adb								X			
MACRO	R. IND								X			
MACRO	R. INT								X			

Description	Identification		Systems	Generator	Autocoder	Autocoder			CT 10K	CT 5K	COBOL
	Word 1	Word 2				AS	RPG	FORTTRAN			
	MACRO	R. RNb						X			
	MACRO	R. TPE						X			
	MACRO	RDLIN			X	X			X	X	
	MACRO	RDSBb			X	X			X	X	
	MACRO	RDSFb			X	X			X	X	
	MACRO	RECYC		X	X	X			X	X	
	MACRO	RINIT	(Used in IBM runs only)								
	MACRO	RIO. A			X						
	MACRO	RIO. B			X						
	MACRO	RIO. C			X						
	MACRO	RIO. D			X						
	MACRO	RIO. E			X						
	MACRO	RLSEb			X	X				X	
	MACRO	RSTRT			X	X			X	X	
	MACRO	RWDbb			X	X			X	X	
	MACRO	S. ONE						X			
	MACRO	S. OPb						X			
	MACRO	S. SLT						X			
	MACRO	S. TWO						X			
	MACRO	SCANb	(Used in IBM runs only)								
	MACRO	SETSW		X	X	X			X	X	
	MACRO	SHIFT		X	X	X			X	X	
	MACRO	SINbb		X	X	X			X	X	
	MACRO	SNAPb		X	X	X			X	X	
	MACRO	SQRTb		X	X	X			X	X	
	MACRO	T. PEB						X			
	MACRO	V. CTR						X			
	MACRO	W. OUT						X			
	MACRO	W. TPE						X			
	MACRO	WSMbb			X	X			X	X	
	MACRO	WTMbb			X	X			X	X	
	MACRO	XCHPA				X					
	MACRO	XCHPB				X					
	MACRO	XCHPC				X					
	MACRO	XCHPD				X					
	MACRO	XCHPE				X					
	MACRO	XCHPF				X					
	MACRO	ZEROb			X	X	X		X	X	
	MACRO	ZSIGN			X	X	X		X	X	
Edit 1	M90bb	P2E1b		X	X	X	X	X	X	X	
Macro Table	M91b*	P2MTA		X	X	X	X	X	X	X	
Edit 2	M95bb	P2E2B		X	X	X	X	X	X	X	

Description	Identification		Systems	Generator	Autocoder	Autocoder AS	RPG	FORTRAN	CT 10K	CT 5K	COBOL
	Word 1	Word 2									
Autocoder Phase I											
Systems Control 6	Q35bb	SYCL6		X	X	X	X		X	X	X
Pass 2											
Coding block 1	Q40bb	P1P2A		X	X	X	X		X	X	X
Coding block 2	Q45bb	P2P2B		X	X	X	X		X	X	X
Pass 3											
Coding block 1	Q50bb	P1P3A		X	X	X	X		X	X	X
Coding block 2	Q55bb	P1P3B		X	X	X	X		X	X	X
Macro Table	Q60b*	P1MTB		X	X	X	X		X	X	X
Autocoder Phase II											
Sort 1	Q70bb	PH2S1		X	X	X	X		X	X	X
Merge 1	Q75bb	PH2M1		X	X	X	X		X	X	X
Information Transfer	Q80bb	P21Tb		X	X	X	X		X	X	X
Macro Table	Q90b*	P2MTB		X	X	X	X		X	X	X
Autocoder Phase III											
Record Construction											
Coding block 1	U15bb	P3RCA		X	X	X	X	X	X	X	X
Coding block 2	U15Ab	P3RCB		X	X	X	X	X	X	X	X
Dump Routine	U21bb	PH3DM	(Used in IBM runs only)								
Availability Table	U30bb	PH3XS		X	X	X	X	X	X	X	X
Sort 1	U30D1	PH3S1		X	X	X	X	X	X	X	X
Merge 1	U30D2	PH3M1		X	X	X	X	X	X	X	X
Serial Transfer	U30Nb	PH3ST		X	X	X	X	X	X	X	X
Sort 2	U31bb	PH3S2		X	X	X	X	X	X	X	X
Merge 2	U32bb	PH3M2		X	X	X	X	X	X	X	X
Assignment Pass											
Coding block 1	U45bb	PH3AS		X	X	X	X	X	X	X	X
Coding block 2	U49bb	PH3AT		X	X	X	X	X	X	X	X
Sort 3	U61bb	PH3S3		X	X	X	X	X	X	X	X
Merge 3	U62bb	PH3M3		X	X	X	X	X	X	X	X
Information Transfer	U68bb	PH3IT		X	X	X	X	X	X	X	X
Sort 4	U71bb	PH3S4		X	X	X	X	X	X	X	X
Merge 4	U72bb	PH3M4		X	X	X	X	X	X	X	X
Output Pass											
Coding block 1	U76bb	P3OPA		X	X	X	X	X	X	X	X
Coding block 2	U77bb	P3OPB		X	X	X	X	X	X	X	X
Overflow Procedure	U771b	PH3OF		X	X	X	X	X	X	X	X
One-for-one Table	U772b	PH3OT		X	X	X	X	X	X	X	X
Commercial Translator 10K											
Systems Control 9	X00bb	SYCL9							X		
Commercial Translator 10K											
Phase IA, Part 2											
System Control	X10bb	1A2CS							X		

Description	Identification		Systems	Generator	Autocoder	Autocoder			CT	CT	COBOL
	Word 1	Word 2				AS	RPG	FORTRAN	10K	5K	
Coding block 1	X10bb	1A2CT							X		
Phase IB											
System Control	X15bb	1BbCS							X		
Coding block 1	X15bb	1BbCT							X		
Phase II											
System Control	X20bb	2bbCS							X		
Coding block 1	X20bb	2bbCT							X		
Phase II-1											
System Control	X25bb	2-1CS							X		
Coding block 1	X25bb	2-1CT							X		
Phase III											
System Control	X30bb	3bbCS							X		
Coding block 1	X30bb	3bbCT							X		
Phase IV											
System Control	X40bb	4bbCS							X		
Coding block 1	X40bb	4bbCS							X		
Coding block 2	X41bb	4bbCT							X		
Coding block 3	X42bb	4bbCT							X		
Phase V											
System Control	X50bb	5bbCS							X		
Coding block 1	X50bb	5bbCT							X		
Coding block 2	X51bb	5bbCT							X		
Phase VI, Part 1											
System Control	X60bb	6P1CS							X		
Coding block 1	X60bb	6P1CT							X		
Coding block 2	X61bb	6P1CT							X		
Phase VI, Part 2											
System Control	X65bb	6P2CS							X		
Coding block 1	X65bb	6P2CT							X		
Phase VII, Part 1											
System Control	X70bb	7P1CS							X		
Coding block 1	X70bb	7P1CT							X		
Coding block 2	X71bb	7P1CT							X		
Phase VII, Part 2											
System Control	X75bb	7P2CS							X		
Coding block 1	X75bb	7P2CT							X		
Coding block 2	X76bb	7P2CT							X		
Phase VIII, Part 1											
System Control	X80bb	8P1CS							X		
Coding block 1	X80bb	8P1CT							X		

Description	Identification		Systems	Generator	Autocoder	Autocoder AS	RPG	FORTRAN	CT 10K	CT 5K	COBOL
	Word 1	Word 2									
Phase VIII, Part 2											
System Control	X85bb	8P2CS							X		
Coding block 1	X85bb	8P2CT							X		
Error Messages	X91bb	MSGb1							X		
FORTRAN											
Systems Control 7	Y15bb	SYCL7						X			
Section 2	Y20bb	FORT2						X			
Section 3	Y30bb	FRT3A						X			
Section 3A	Y35bb	FRT3B						X			
Section 4	Y40bb	FORT4						X			
Section 5	Y50bb	FORT5						X			
Section 6	Y60bb	FORT6						X			
Section 7	Y70bb	FRT7A						X			
Section 7A	Y75bb	FRT7B						X			
Section 7B	Y79bb	FRT7C						X			
Report Generator											
Systems Control 8	215bb	SYCL8					X				
Phase 2											
Sort 1	221bb	RPG21					X				
Merge 1	222bb	RPG22					X				
Phase 3	231bb	RPG31					X				
Phase 4											
Sort 2	241bb	RPG41					X				
Merge 2	242bb	RPG42					X				
Phase 5	251bb	RPG51					X				
Phase 6											
Pass 1	261bb	RPG61					X				
Pass 2	262bb	RPG62					X				
Pass 3	263bb	RPG63					X				
Phase 7	271bb	RPG71					X				
COBOL											
Systems Control 11	500bb	SYC11									X
Phase IB											
System Control	511Bb	1Bb1S									X
Coding block 1	511Bb	1Bb1T									X
Phase IC											
System Control	511Cb	1Cb1S									X
Coding block 1	511Cb	1Cb1T									X
Phase IIA											
System Control	512Ab	2Ab1S									X
Coding block 1	512Ab	2Ab1T									X

Description	Identification		Systems	Generator	Autocoder	Autocoder		RPG	FORTRAN	CT	CT	COBOL
	Word 1	Word 2				AS	10K			5K		
Phase IIB												
System Control	512Bb	2Bb1S										X
Coding block 1	512Bb	2Bb1T										X
Phase IIC												
System Control	512Cb	2Cb1S										X
Coding block 1	512Cb	2Cb1T										X
Phase III												
System Control	513bb	3bb1S										X
Coding block 1	513bb	3bb1T										X
Phase IV												
System Control	514bb	4bb1S										X
Coding block 1	514b1	4b11T										X
Coding block 2	514b2	4b21T										X
Phase V												
System Control	515bb	5bb1S										X
Coding block 1	515b1	5b611T										X
Coding block 2	515b2	5b21T										X
Phase VIA												
System Control	516Ab	6Ab1S										X
Coding block 1	516A1	6A11T										X
Coding block 2	516A2	6A21T										X
Phase VIB												
System Control	516Bb	6Bb1S										X
Coding block 1	516Bb	6Bb1T										X
Phase VIIA												
System Control	517Ab	7Ab1S										X
Coding block 1	517A1	7A11T										X
Coding block 2	517A2	7A21T										X
Phase VIIB												
System Control	517Bb	7Bb1S										X
Coding block 1	517B1	7B11T										X
Coding block 2	517B2	7B21T										X
Phase VIIIA												
System Control	518Ab	8Ab1S										X
Coding block 1	518Ab	8Ab1T										X
Phase VIIBB												
System Control	518Bb	8Bb1S										X
Coding block 1	518B1	8B11T										X
Coding block 2	518B2	8B21T										X
Commercial Translator - 5K												
Phase 1D												
System Control	571Db	1Db7S									X	
Coding block	571Db	1Db7T									X	

<u>Description</u>	<u>Identification</u>		<u>Systems</u>	<u>Generator</u>	<u>Autocoder</u>	<u>Autocoder</u>			<u>CT</u>	<u>CT</u>	<u>COBOL</u>
	<u>Word 1</u>	<u>Word 2</u>				<u>AS</u>	<u>RPG</u>	<u>FORTTRAN</u>	<u>10K</u>	<u>5K</u>	
Phase 1E											
System Control	571Eb	1Eb7S									X
Coding block 1	571Eb	1Eb7T									X
Phase 2A											
System Control	572Ab	2Ab7S									X
Coding block	572Ab	2Ab7T									X
Phase 2B	572Bb	2Bb7T									X
Phase 2C											
System Control	572Cb	2Cb7S									X
Coding block 1	572Cb	2Cb7T									X
Phase 3A											
System Control	573Ab	3Ab7S									X
Coding block 1	573Ab	3Ab7T									X
Phase 3B											
System Control	573Bb	3Bb7S									X
Coding block 1	573Bb	3Bb7T									X
Phase 3C	573Cb	3Cb7T									X
Phase 4A											
System Control	574Ab	4Ab7S									X
Coding block 1	574Ab	4Ab7T									X
Phase 4B	574Bb	4Bb7T									X
Phase 4C	574Cb	4Cb7T									X
Phase 4D	574Db	4Db7T									X
Phase 4E	574Eb	4Eb7T									X
Phase 4F	574Fb	4Fb7T									X
Phase 4G	574Gb	4Gb7T									X
Phase 5A											
System Control	575Ab	5Ab7S									X
Coding block 1	575Ab	5Ab7T									X
Phase 5B	575Bb	5Bb7T									X
Phase 6A											
System Control	576Ab	6Ab7S									X
Coding block 1	576Ab	6Ab7T									X
Phase 6B	576Bb	6Bb7T									X
Phase 6C											
System Control	576Cb	6Cb7S									X
Coding block 1	576Cb	6Cb7T									X
Phase 7A											
System Control	577Ab	7Ab7S									X
Coding block 1	577Ab	7Ab7T									X
Phase 7B	577Bb	7Bb7T									X

<u>Description</u>	<u>Identification</u>		<u>Systems</u>	<u>Generator</u>	<u>Autocoder</u>	<u>Autocoder</u>			<u>CT</u>	<u>CT</u>	<u>COBOL</u>
	<u>Word 1</u>	<u>Word 2</u>				<u>AS</u>	<u>RPG</u>	<u>FORTTRAN</u>	<u>10K</u>	<u>5K</u>	
Phase 7C											
System Control	577Cb	7Cb7S									X
Coding block 1	577Cb	7Cb7T									X
Phase 7D	577Db	7Db7T									X
Phase 8A											
System Control	578Ab	8Ab7S									X
Coding block 1	578Ab	8Ab7T									X
Phase 8B											
System Control	578Bb	8Bb7S									X
Coding block 1	578Bb	8Bb7T									X
Error Messages	578Cb	8Cb7T									X
Tape Mark	99999	99999									

APPENDIX E: COMMUNICATIONS RECORD

The Communications Record is a section of the Compiler Systems Tape where a record is maintained of the information required by the processor for a run. The permanent operating options included in the systems tape are specified in the record. All temporary and permanent changes in the operating options are recorded automatically as a result of reading an operating option control card. A RUN Control Card will cause the type of run to be recorded. The version and change level of the systems

tape are part of the record, and are used in checking the tape for an official IBM patch or addition.

The processor refers to the Communications Record to determine the type of run to be made and the equipment to be used.

The DA which reserves and defines the Communications Record appears on this page. The area of storage assigned to each of the entries is indicated.

LN	CDREF	LABEL	OP	OPERAND	CDNO	FD	LOC
01	0101	ORIGIN	CNTRL	130			
02	0102	COMREC	DA	1			
03	0104	MAXMEMORY		00, 04		04	0130
04	0105	MINMEMORY		05, 09		59	0130
05	0106	LOADPUNCH		10, 11		01	0131
06	0107	PRINTLINES		12, 13		23	0131
07	01071	OPERATIONS		14		44	0131
08	0108	STARTLOCN		15, 19		59	0131
09	0109	FLOATDECM		20		00	0132
10	0110	LOADMODE		21		11	0132
11	0111	PUNCHDECK		22		22	0132
12	0112	PRINTLIST		23		33	0132
13	0113	LOADPROG		24		44	0132
14	01131	LOADMODEF		25		55	0132
15	0114	HIGHMEMORY2		26, 29		69	0132
16	0115	CHANNELS		30		00	0133
17	0116	HIGHMEMORY		31, 34		14	0133
18	0117	PUNCHSYNC		35		55	0133
19	0118	PRINTSYNC		36		66	0133
20	01181	ERRORPRINT		37		77	0133
21	01182	SEVNTHUNIT		38, 39		89	0133
22	0119	TAPESET1		40, 49		09	0134
23	0120	TAPESET2		50, 59		09	0135
24	0121	INITAUNIT		60, 61		01	0136
25	0122	INTBUNIT		62, 63		23	0136
26	0123	MAINUNIT		64, 65		45	0136
27	0124	INLABELS		66		66	0136
28	0125	INLABELINF		67		77	0136
29	0126	INDENSITY		68		88	0136
30	0127	INTDENSITY		69		99	0136
31	0128	WORKLABELS		70		00	0137
32	0129	WKTENSITY		71		11	0137
33	0130	OUTLABELS		72		22	0137
34	0201	OUTLABLINF		73		33	0137
35	0202	OUTDENSITY		74		44	0137
36	0203	OUTTDENSTY		75		55	0137
37	0204	OUTOPENPRC		76		66	0137
38	02041	PROCESSADD		77		77	0137
39	02042	ADDSTORAGE		78		88	0137
40	02043	POSTPROC		79		99	0137
41	0205	DECKTAPE		80, 81		01	0138
42	0206	LISTAPE		82, 83		23	0138
43	0207	PH3DECK		84, 85		45	0138
44	0208	PH3LIST		86, 87		67	0138
45	02081	LISTBLOCK		88		88	0138
46	0209	COMPLETE		89		99	0138
47	0210	MULTIFILE		90		00	0139
48	0211	RUNTYPE		91		11	0139
49	0212	MACROGEN		92		22	0139
50	0213	MACROLEVEL		93, 94		34	0139
51	0214	RPGIND		95		55	0139
52	0215	MAINSAVE		96, 97		67	0139
53	0216	INDENSAVE		98		88	0139
54	0217	IOFIELD		99		99	0139
55	0218	MACRONAME		100, 109		09	0140
56	0219	MACROTYPE		110, 119		09	0141
57	02191	RECOMP		120		00	0142
58	02192	CHANGEUNIT		121, 122		12	0142
59	02193	LOADPUNCHF		123, 124		34	0142
60	02194	MIXEDTAPE		125		55	0142
61	0220	VERSION		180, 189		09	0148
62	0221	CHNGLEVEL		190, 194		04	0149

APPENDIX F: GLOSSARY

ALPHAMERIC

A term which refers to symbols that are numerical digits, alphabetic characters, or special characters.

7070 AUTOCODER

Autocoder is a symbolic programming system consisting of a symbolic language and a compiler. The system produces a machine-language object program for a 7070, 7072, or a 7074 from a source program written in the Autocoder language.

7070 COBOL

7070 COBOL is a symbolic programming system consisting of the COBOL language and the COBOL processor. The language, which closely resembles English, is essentially problem-oriented rather than machine-oriented. The system produces a program in 7070 Autocoder language from a source program coded in COBOL language.

COLLATING SEQUENCE

The relative order of precedence which a computer assigns to the numbers, letters, and special characters for compare operations.

7070 COMMERCIAL TRANSLATOR

7070 Commercial Translator is a symbolic programming system consisting of the Commercial Translator language and the Commercial Translator processor. The language, which closely resembles English, is essentially problem-oriented rather than machine-oriented. The system produces a program in 7070 Autocoder language from a source program coded in Commercial Translator language.

COMPILE

To produce a machine-language routine from a routine written in non-machine language. See also: COMPILER

COMPILER

A compiler is a complex program in which several different functions are performed. It typically includes the following:

1. Extensive program analysis during which information is collected or tabulated for later recombination.

2. Generation of instructions by synthesis of tabulated information and use of skeleton or model routines.

3. Translation of symbolic instructions into machine language.

A compiler is itself a routine, not a machine - although a machine could be built to do compiling.

CONDENSED CARD FORMAT

This is a format for placing several machine language instructions on a single card with their designated location.

CONTROL CARD

A card which contains the parameters required to set up a generalized program for one particular application.

DEBUGGING

The process of locating errors in a computer routine and correcting them.

FILE

An organized collection of information directed toward some purpose.

7070 FORTRAN

7070 FORTRAN is symbolic programming system consisting of the FORTRAN language and the 7070 FORTRAN compiler. The language, which closely resembles the language of mathematics, is essentially problem-oriented rather than machine-oriented. The system produces a program in 7070 Autocoder language from a source program coded in the FORTRAN language. The Autocoder program is converted to a machine-language object program for a 7070 or a 7074 by the Autocoder processor.

GENERATOR

A program which accepts input parameters and uses them to modify skeleton instructions or skeleton routines to produce the desired output routine. A small number of parameters are capable of producing a large number of output instructions.

HASH TOTAL

A total of data, for auditing or control purposes, which would not ordinarily be added together, e.g.,

summing a list of parts numbers. In multi-pass computer applications (e.g., tape sorting), such hash totals are compared at the end of each pass with the hash total from the preceding pass. This provides a check for machine or program failures and may detect the presence of illegal characters in data.

HEADER LABEL

A record at the beginning of a tape for identification and control. The header label usually contains file identification, creation date, retention cycle, and indication of switch settings and manual operations for the console operator.

LABEL

A record magnetically written on a tape to identify its contents. See also: TAPE LABEL. Also, a paper label attached to a reel of magnetic tape to identify its contents. In Autocoder, columns 6-15 of a control card or Autocoder Input Card.

LOAD

To transfer the instructions composing a routine from some external storage device into the primary storage of a computer where the instructions may be executed. Loading usually must be initiated by an operator action, but is generally accomplished by a load routine which itself is in core storage. See also: CONDENSED CARD FORMAT.

LOG

A detailed record of machine and program operation.

MACHINE-LANGUAGE CODING

Coding of instructions in a form which may be executed by the computer without translation.

MACRO GENERATOR

A macro generator, an abbreviation of macro-instruction generator, produces the sequence of symbolic instructions which is referenced by a macro-instruction.

MACRO-INSTRUCTION

A symbolically coded instruction resulting in a group of machine-language instructions which will perform a desired operation.

OBJECT MACHINE

The machine on which an object program is to be run. See: PROCESSOR MACHINE.

OBJECT PROGRAM

The output from a processor, in this case, a 7070/7072/7074 machine-language program assembled from a source program coded in symbolic language.

OBJECT TIME

The output from a processor, in this case, a 7070/7072/7074 machine-language program assembled from a source program coded in symbolic language.

OFF-LINE

Operation of input/output and other devices not under direct computer control. Most commonly used to designate the transfer of information between magnetic tapes and other input/output media.

ON-LINE

Operation of an input/output device as a component of the computer, under programmed control.

PARAMETER

A factor which is left unspecified and to which the user may assign a value.

PROCESS TIME

The time at which the source program is being changed into an object program by a compiler, such as Autocoder. This is opposed to object time, the time at which the object program is being run.

PROCESSOR

A program which performs the functions of assembly, compilation, generation, or any similar functions.

PROCESSOR MACHINE

The machine on which a processor is to be run.

REPORT

A printed document which presents data arranged in an orderly manner for ease of reference.

REPORT GENERATION

A technique for producing complete reports given the content and format of the input file and the desired content and format of the output reports.

7070 REPORT PROGRAM GENERATOR

The 7070 Report Program Generator consists of coding forms and a compiler. The format of the input file and the specifications for the output report are placed on the coding sheets which are then punched into cards and used as input for the compiler. The system produces a program in 7070 Autocoder language which is converted to a machine-language object program for a 7070, 7072, or 7074 by the Autocoder processor.

RUN

A general term used to designate the operation of a program or a routine for some specific purpose. Runs discussed in this manual, such as Compile Run and Generator Run, are runs of a specific type. Runs, such as payroll runs and inventory runs, are referred to in general terms by the type of function they perform.

SORT

To place a file of records in order according to a specified sequence.

SOURCE LANGUAGE

The language in which a program is coded, e. g., Autocoder.

SOURCE PROGRAM

The original coding of a program, usually coding in a language other than machine language.

STORAGE

Any medium into which data may be transferred and where it may be retained for later use.

SYMBOLIC-LANGUAGE CODING

Coding of instructions in a mnemonic language, such as Autocoder, rather than in machine language. Symbolic-language coding must be translated into machine language to be used by the computer.

TAPE LABEL

A record at the beginning and/or end of a tape for identification and control. See: HEADER LABEL, TRAILER LABEL.

TRAILER LABEL

A record at the end of a tape for identification and control. The trailer label usually contains the record count, control totals, and end-of-reel notations.

INDEX

(Where more than one page reference is given for a particular subject, the major reference is underscored.)

- Attributes Word 12-13
- Checkpoint, COBOL and Commercial Translator 42
- Coding Blocks on Systems Tape 51-62
- Communications Record 63
- Compile Run 2, 6-8
 - Description 6
 - Halts 8
 - Input/Output Configurations 7
 - Operating Procedure 6-8
 - RUN Control Cards 6, 19
- Composition, Compiler Systems Tape 51-62
- Control Cards, Operating Option 2, 19-28
 - ADDSTORAGE 23
 - CHANNELS 21
 - DECKTAPE 22
 - ERRORPRINT 26
 - FLOATDECML 24
 - HIGHMEMORY 21
 - INDENSITY 21
 - INITAUNIT 27
 - INITBUNIT 27-28
 - INLABELS 21-22
 - LISTBLOCK 25-26
 - LISTTAPE 22
 - LOADMODE 24
 - LOADMODEF 24
 - LOADPROG 23
 - LOADPUNCH 25
 - LOADPUNCHF 25
 - MAINUNIT 21
 - MAXMEMORY 23
 - MINMEMORY 23
 - MIXEDTAPE 26
 - OUTDENSITY 22
 - OUTLABELS 22-23
 - PRINTLINES 24
 - PRINTLIST 25
 - PRINTSYNC 21
 - SEVNTHUNIT 23
 - SRELEASE 27
 - SRESERVE 27
 - STARTLOCN 24
 - WORKLABELS 22
 - XRELEASE 27
 - XRESERVE 26
- Control Cards, RUN 2
 - COMPILE 6, 19
 - GENERATOR 12, 19
 - MULTIFILE 16, 19
 - SYSTEMS 9, 19
- Control Cards, Systems Update 9-10, 28
 - DROP 9, 28
 - ENTER 10, 28
 - PATCH 10, 28
- Control Words, FORTRAN 45-47
- DECKTAPE 15
- Execute Cards 44
- Formats, Input/Output 3
- FORTRAN Object Program, Loading 43-49
- Generator Run 2, 12-15
 - Description 12
 - Halts 15
 - Input/Output Configurations 14
 - Operating Procedure 13-15
 - RUN Control Card 12
- Glossary 64-66
- Halts and Messages
 - FORTRAN Loader 47-49
 - Programmed 30-37
 - INITAUNIT 15, 27
 - INITBUNIT 15, 27-28
 - Initial Input 4
 - Input Options 4
 - Initial Input 4
 - Main Input 4
 - Input/Output Formats 3
 - Input/Output Unit Allocation 4-5
 - Input/Output Unit Requirements 4
 - Installation and Operation Suggestions 27
 - LISTTAPE 15
 - Loading FORTRAN Object Program 43-49
 - Macro Generators 12-13
 - Main Input 4
 - MAINUNIT 15
 - Messages
 - Compiler Systems Tape 38-41
 - FORTRAN Loader Halts and 47-49
 - Optional Typed 5
 - Programmed Halts and 30-37
- Multifile Run 2, 15-18
 - Description 15
 - Halts 18
 - Input/Output Configurations 17
 - Operating Procedure 16-18
 - RUN Control Card 16, 19
- Object Machine 2, 23-27
- Operating Option Control Cards 2, 19-28
 - List of 20
 - Object Machine 23-27
 - Permanent Operating Option Changes 26-28
 - Processor Machine 21-23
- Operating Options 2
- Operating Procedure
 - Compile Run 6-8
 - FORTRAN Object Program 47
 - Generator Run 13-15
 - Multifile Run 16-18
 - Report Program 50
 - Systems Run 10-12
- Operation and Installation Suggestions 29
- Output Options 4
- Processor Machine 2, 21-23
- Programmed Halts and Messages 30-37
- Relocatable Loader, FORTRAN 43
- Relocatable Routines, Requirements of 43
- Report Program Generator Input Sequence 6
- Report Program Operating Procedures 50
- RUN Control Cards (see Control Cards, RUN)
- Run Types 2
 - Compile Run 2, 6-8
 - Generator Run 2, 12-15
 - Multifile Run 2, 15-18
 - Systems Run 2, 8-12
- Systems Run 2, 8-12
 - Description 8
 - Halts 12
 - Input/Output Configurations 11
 - Operating Procedure 10-12
 - RUN Control Cards 9, 19
 - Systems Update Control Cards 9-10, 28
- Systems Update Control Cards (see Control Cards, Systems Update)
- Title Cards 43
- Transfer Entry Cards 44

IBM

**International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York**