File No. 1410-27
Form C28-0287-1

IBM Systems Reference Library

IBM 1410 Processor Operating System

Using IBM 1301 Disk Storage

This publication describes the functions and use of the IBM 1410 Processor Operating System using IBM 1301 Disk Storage.

The first section, intended primarily for programmers, system analysts, and supervisory personnel, describes the components of the operating system, operating configurations, control card formats, card deck arrangements, programming requirements, and operating-system features and limitations.

The second section, intended for machine room personnel, describes step-by-step operating procedures. It also contains lists of halts, messages, and appropriate corrective actions. Appendices describe the off-line print/punch program that can be used with the system, important internal control tables, and the FORTRAN Relocatable-Program Loader.

MAJOR REVISION

This publication supersedes and obsoltes the publication, IBM 1410 Processor Operating System Using IBM 1301 Disk Storage, Form C28-0287. New information concerning FORTRAN, COBOL, and the Disk Update Program is included in this publication, as well as additions and corrections to the material previously published in Form C28-0287. A new appendix, describing the FORTRAN Relocatable-Program Loader is also included.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the content of this publication to:
IBM Corporation, Programming Systems Publications, Dept. D91, PO Box 390, Poughkeepsie, N. Y.

CONTENTS

INTRODUCTION 5
Purpose of this Publication 5
Purpose of POS 5
Prerequisites 5
  To Use POS 5
  To Write Source Programs 5
Machine Requirements 5
  Options 6

DETAILED MACHINE REQUIREMENTS AND CONSIDERATIONS 7
Cylinder Requirements 7
Format Track Requirements 7
Use of Additional Cylinders to Expand System Functions . . 8
Use of Additional Core Storage to Expand System Functions . 8

GENERAL CONSIDERATIONS -- POS 9
System Programs 9
  System Supervisory Programs 9
  System Library 9
  Processors 9

CREATION OF POS -- THE BUILD RUN 10
General 10
  Card Oriented Systems 10
Control Cards 10
  CREAT Cards 10
  CONFG Cards 11
  Standard Operating Configuration on System Tape Supplied. 14
The Build Run Deck 15

PROCESSOR RUNS 16
General 16
Control Cards 16
  RUN Card 16
  EOJ Card 16
Autocoder Considerations 16
  General 16
  END Cards 16
  CTL Cards 17
  LOAD Cards 17
  Restrictions 17
  Autocoder Run Deck Setup 17
RPG Considerations 17
  General 17
  Assembly Option 18
  Suppression of Punch and/or Print 18
  RPG Run Deck Setup 18
COBOL Considerations 18
  General 18
  COBOL Run Options 18
FORTRAN Considerations 19
  General 19
  Batch Compiling 19

MAINTENANCE AND AUXILIARY FUNCTIONS -- UPDATE AND PRINT RUNS 20
General 20
The Update Run 20
  Definition Cards 20
  Updating a Phase 21
  Updating a Program 22
  Listing the Library from the System Tape 23
  Changing the Standard Operating Configuration 23
  Updating the Library 23
  Duplicating System Tape 24
  Card Deck Set for the Update Run 24
1301 Print Run 24
  Control Card. 24
  Card Deck Setup for the Print Run 25
Disk Update Program 25
  General 25
  Control Cards 25

APPENDIX A -- THE 1401 PRINT/PUNCH PROGRAM . . . 27
  What this Program Will Do 27
  Machine Requirements 27
  Control-Card and Sense-Switch Options 27
  Operator's Guide 27
  Error Correction. 28

APPENDIX B -- CONTROL TABLES 29
The Communication Table 29
Program and Phase Directory 29
  Program Entries 29
  Phase Entries. 29
The Library Directory. 29

APPENDIX C -- FORTRAN RELOCATABLE-PROGRAM LOADER 31
Introduction to Relocation 31
FORTRAN Relocatable-Program Loader 31
Autocoder Relocation 31
Relocation Indicators. 33
Control Cards 34
Library Tape Unit 34
Enter From 34
Final Input Tape 34
Lower Load Limit 35
Upper Load Limit 35
Intermediate Tape Unit 35
Replace Printer By. 35
Object Tape Assignment. 35
Module Available 1301 36
Channel Number 1301 36
Home Address 36
Memory Map 36
Write Relocated Tape 36
Punch Relocated Deck 37
Clear Memory From 37
Sense Switch Settings. 37

PROGRAMMING CONSIDERATIONS 38
Logic Description 38
Program Assembly Error Conditions 40

LIBRARY 41
Subroutines 41
Inclusion of User Routines at POS Creation 42

OPERATING INSTRUCTIONS 43
Messages 43

OPERATOR'S GUIDE 45
The Start Instruction 45
Build Run 45
Tape Update Run 45
Bootstrap Routine 46
Processor, Print, and Disk Update Runs 46
Compound Runs 46
Halts, Messages, and Corrective Actions 47
Identifying Message Numbers 47
Halts, Messages, and Corrective Action for POS Runs . . 47
COBOL Processor Run Messages (Console) 56
FORTRAN Processor Run Messages (Console) 56
Halts, Messages, and Corrective Actions for Object Program Runs 57
COBOL Object Program Messages 64
FORTRAN Object Program Messages 64

INDEX 65

PURPOSE OF THIS PUBLICATION

This publication contains information that enables an installation to use and maintain the "1410 Processor Operating System using 1301 Disk Storage." Also included is material that is of specific interest to programmers and system analysts. (The "1410 Processor Operating System using 1301 Disk Storage" will hereafter be referred to as "POS.")

PURPOSE OF POS

POS comprises an integrated set of programs that produce machine-language object programs for the 1410 Data Processing System from source programs written in the Autocoder, Report Program Generator (RPG), COBOL or FORTRAN languages. The random-access capabilities of 1301 Disk Storage are used both for storage of the system and for intermediate work areas.

Additional Features and Advantages

POS also provides the following features and advantages:

1. Additions, deletions, and corrections can easily be made to the system.
2. The user can make additions to the POS Library of macros and sub-routines.
3. Selected sections of the POS Library can be printed out.
4. If the POS is on tape, it can be duplicated.
5. A uniform IOCS is provided for all programs in the system.
6. All processors for the 1410 Data Processing System are stored on a single 1301 Disk Storage unit. All processors are distributed on one reel of magnetic tape, the System Tape.
7. The user can operate within a wide range of machine configurations. Each installation can define its own standard operating configuration.
8. Operating procedures are simple and operator intervention is minimum.

PREREQUISITES

To Use POS

The user should be familiar with the material contained in the following reference manuals:

"IBM 1410 Principles of Operation," Form A22-0526
"IBM 1301, Models 1 and 2, Disk Storage and IBM 1302, Models 1 and 2, Disk Storage with IBM 1410 and 7010 Data Processing Systems," Form A22-6788

To Write Source Programs

The user should consult the following publications for information on the writing of programs in the source language of each processor:

"IBM 1410 Autocoder," Form C28-0309
"Report Program Generator for the IBM 1410," Form C28-1443
"Report Program Generator for the IBM 1410 — Supplement for 1301 Disk Storage," Form J28-0256
"IBM 1410 Programming Systems; COBOL — Additional Specifications," Form J28-0232
"IBM 1410 FORTRAN," Form J24-1468

The following publications should be consulted for details on IOCS:

"IBM 1410 Input/Output Control System for Card and Tape Systems," Form C28-0334
"IBM 1410 Input/Output Control System for 1301 Disk Storage," Form C28-0251
"IBM 1410 Input/Output Control System for 1405 Disk Storage," Form J28-0233

This publication does not discuss any of the material included in the above publications. It does, however, specify restrictions (if any) imposed by POS that must be observed when writing source programs.

MACHINE REQUIREMENTS

The following are the minimum machine requirements for using POS:

40,000 positions of core storage
Processing Overlap and Priority special features
1 IBM 1402 Card Read-Punch, Model 2
1 IBM 1403 Printer, Model 2
40 cylinders of 1301 Disk Storage (see "Detailed Machine Requirements and Considerations").

NOTE: Two tape units are required by the Update Program. (If control card parameters are on tape, a third tape unit is required.)

Options

One IBM 729 II, IV, V, VI or 7330 Magnetic Tape Unit can fill each of the functions of card reading, card punching, and printing.

Data for punch and print output can be intermixed on one reel of magnetic tape for subsequent off-line printing on an IBM 1401 Data Processing System. If this option is chosen, the data is automatically preceded by a self-loading 1401 program that selectively prints and punches.

CYLINDER REQUIREMENTS

POS, as supplied by IBM, requires a minimum of 40 cylinders of disk storage, grouped and distributed as described below.

The user must specify four separate groups of cylinders (see the section, "Creation of POS -- The Build Run"). Each group is called a System Area, or simply, an area. A minimum of five consecutive load-mode cylinders on one module of disk storage is needed as System Area 1 for the storage of the mandatory programs that comprise POS and any load-mode work area required by the system programs. The remaining 35 cylinders are move-mode areas used for working storage and intermediate processing.

The number of tracks required by each program in the POS system are shown in Table 1; the table specifies the mandatory and optional track requirements for the POS system programs.

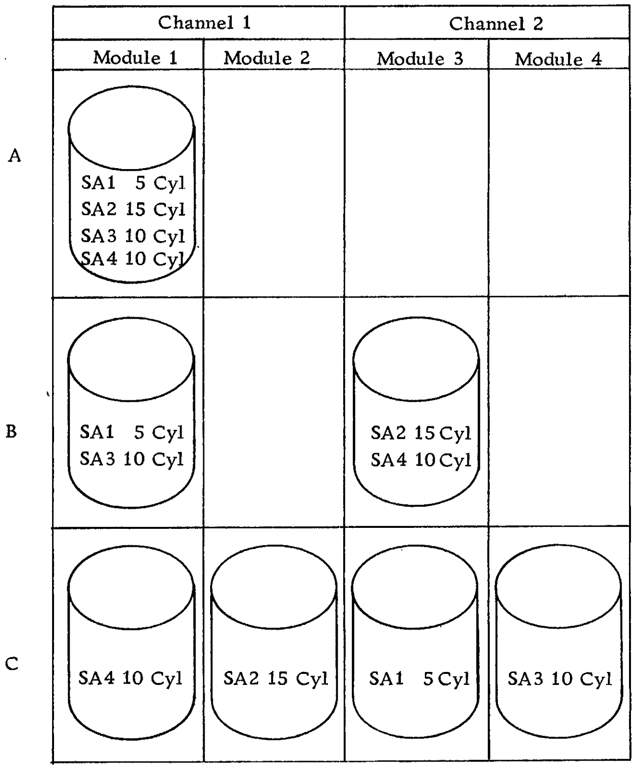
It should be pointed out that whenever the IOCS or COBOL library is requested, it begins on the first track of the next cylinder after the POS system programs. For example, if the last track used by the last system program is 5231 (CYL 130; TRK 32), the first track used by the Library is 5240 (CYL 131; TRK 1).

System Program	Tracks Required	Mandatory	Optional
System Supervisory Programs	25	x	
Autocoder	32	x	
FORTRAN	61		x
COBOL	96		x
RPG	78		x
PRINT	2		x
IOCS library	132		x
COBOL Library	29		x
Working Storage	120	x	

Table 1. Disk Track Requirements

The 35 cylinders used for working storage and intermediate processing are specified in three additional System Areas. These must be specified as areas of 15, 10 and 10 cylinders each. These three areas may be placed on three additional disk modules, making a possible maximum total of four modules for POS, on either or both channels.

The four areas can be defined for a single module of 1301 Disk Storage. However, POS will operate most efficiently if the four areas are distributed on different modules and over two channels. The two areas of 10 cylinders each should be placed, whenever possible, on different modules. The four areas may be on contiguous cylinders in the same module, but the user is always required to specify each of the four areas. Figure 1 gives a few examples of possible distributions of the System Areas.



System Area 1 (SA1) = POS Storage--5 Cylinders
System Areas 2, 3, 4 (SA2, 3, 4) = Working Storage--35 Cylinders

Figure 1. Examples of Distribution of System Area

FORMAT TRACK REQUIREMENTS

The format tracks for all POS cylinders must be defined as full tracks, i.e., one record for each full track. The cylinders that are used for storage of POS must be set to the LOAD mode (2,165 characters per track) before POS is loaded into disk storage.

The 35 cylinders of working storage must be set to the MOVE mode (2,800 characters per track) before a source program is processed.

To set the format tracks, it is suggested that the user employ the Format Track Generation and Home Address and Record Address Generation programs described in "Utility Programs for 1301 Disk Storage," Form C28-0279. If these programs are used, the home address and record address must be the same as specified in the System Area CONFG cards specified in this publication.

NOTE 1: If format tracks are not set correctly, unpredictable errors can occur (e.g., looping, invalid output, and unexplained halts).

NOTE 2: The Format Track Generation and Home Address and Record Address Generation programs, as supplied, place "00" in the HA2 fields.

USE OF ADDITIONAL CYLINDERS TO EXPAND SYSTEM FUNCTIONS

An installation may require more disk-storage cylinders than are required by the POS supplied by IBM.

The minimum requirement for System Area 1 is five cylinders. To ensure that System Area 1 can accommodate all system programs while retaining its ability to assemble large source programs, however, the use of 15 cylinders is recommended. Conditions that may arise during program assembly if the capacity of System Area 1 is exceeded are listed in the "Programming Considerations" section of this manual under "Program Assembly Error Conditions."

In addition, if the user adds subroutines to the System Library, one additional cylinder for System Area 1 should be allowed for each 1,000 statements added. For example, if 15 cylinders have been assigned for System Area 1 and 2,000 statements are to be added to the System Library, the allocation for System Area 1 should be increased to 17 cylinders.

The 35 cylinders designated as working storage (System Areas 2, 3, and 4) are normal for creation of object programs that produce up to 8,400 lines of print on the autocoder listing (equivalent to object programs requiring approximately 40,000 to 80,000 core-storage positions). If the user's program exceeds this range, one additional cylinder for each work area must be allowed for each 4,000 positions needed above this range; however, when additional cylinders are added to work areas, one additional cylinder must be specified for each of the three work areas. Referring to Figure 1, if the user requires 4,000 positions above the specified range, System Areas 2, 3, and 4 should be defined to contain 16, 11, and 11 cylinders, making a total of 38 cylinders.

USE OF ADDITIONAL CORE STORAGE TO EXPAND SYSTEM FUNCTIONS

POS requires that the 1410 configuration have a minimum of 40,000 positions of core storage. If there are more positions available, the Autocoder and COBOL processors will take advantage of this increased capability by reserving larger areas for tables. Processing time, therefore, may be reduced for these two processors.

SYSTEM PROGRAMS

The term "System Program" refers to any of the programs forming the Processor Operating System (Figure 2). These programs are described below.

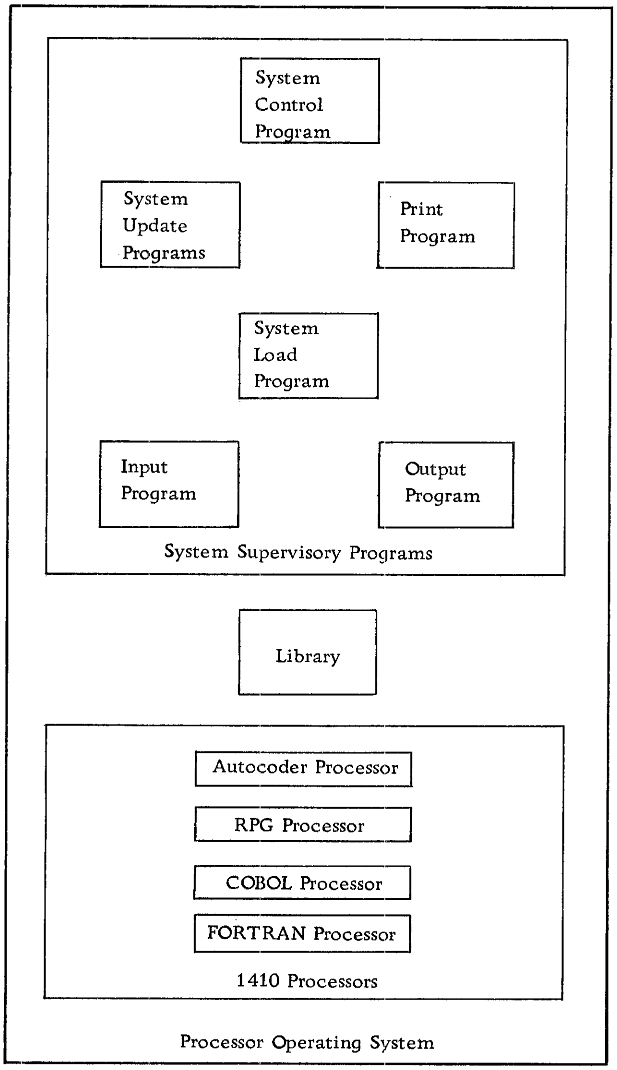


Figure 2. The Processor Operating System

System Supervisory Programs

The System Supervisory Programs control the various functions of POS. There are seven System Supervisory Programs: System Control Program, System Load Program, Tape Update Program, Disk Update Program, Print Program, Input Program and Output Program.

System Control Program: The System Control Program is the central control program of POS. Specifically, under the direction of control cards supplied by the user, it selects processors, performs error checking, provides transition between the various phases of POS, checks tape labels, provides IOCS for all other phases, and determines tape assignments.

System Load Program: The System Load Program, under the direction of control cards supplied by the user, selects and places into disk storage the system programs of POS.

Tape Update Program: The Tape Update Program contains routines that update the system programs. It is used to make corrections, additions, and deletions to the POS System Tape. This program requires that the POS be on the System Tape.

Print Program: The Print Program, under the direction of control cards, prints out selected sections of the System Library.

Input Program: This program reads in all input data to be processed by the individual system programs.

Output Program: This program performs all output functions for all system programs except the Print Program.

Disk Update Program: This program performs the functions required for the maintenance of POS on the disk itself. In many respects it is similar to the Tape Update Program.

System Library

The System Library consists of IOCS subroutines and COBOL subroutines. The user can also add his own subroutines.

Processors

A processor is a machine-language program that performs all the functions necessary to transform a source program into the required object program. The following processors are System Programs in POS:

A. Autocoder C. COBOL
B. RPG D. FORTRAN

Note that the final phase of the RPG, COBOL, and FORTRAN processors consists of program assembly and macro-expansion by the Autocoder Processor.

CREATION OF POS -- THE BUILD RUN

GENERAL

As supplied by IBM, all system programs are on a single reel of magnetic tape -- the System Tape -- which is divided into two files. The first file is in card-image form, and the second is blocked three card records to one. The first file contains the Process Selection Phase, Tape Update Program, Load Disk, and the CONFG and CREAT card package. The second file contains the remainder of the System phases and the Library.

The Build Run is used to create a POS for the user's particular requirements. The user places into disk storage only those system programs that he selects from the System Tape. For example, a commercial installation may not use FORTRAN, but may require the other three processors. The user indicates, through control-card entries, that he requires only the Autocoder, RPG and COBOL processors. The Sytem Load Program builds a POS with these three processors and omits the FORTRAN processor. These choices are indicated to the System Load Program by means of "CREAT" control-card entries.

The user can also set up his standard operating configuration during the Build Run. The parameters which determine the standard operating configuration are specified through CONFG control-card entries.

Standard Operating Configuration

By means of the Build Run, the user can specify parameters that set up the standard operating configuration for his installation. These specify such things as the unit from which his source language programs will be read during a processor run, the location of other input/output devices, whether or not header labels are to be checked, the size of the object machine, and whether the object program is to be written in relocatable form.

During a processor, maintenance or auxiliary run, the user simply places his input information into a preassigned unit and keys in the proper start instructions.

POS allows the user to change the standard operating configuration for any run simply by use of the same type of control cards that were used to set up the standard operating configuration. For example, if the standard operating configuration requires that all source-language input data is to be read in from tape unit number 8 on channel 1, and a particular source language program is on cards, then the user can temporarily change the standard operating configuration by specifying that the source input deck be read in from the card reader for this one run by means of control cards included with this source-language program.

NOTE 1: The System Tape supplied by IBM also contains parameters that determine a standard operating configuration. See "Standard Operating Configuration on System Tape Supplied" in this section of the manual.

NOTE 2: The user can also specify the standard operating configuration by means of the Update Run (see the section, "Update Run").

Card Oriented Systems

If an installation does not have a magnetic tape unit attached to its 1410 system, the System Tape data must be transferred into cards by means of peripheral equipment. The user then inserts the control cards described below into the deck and proceeds as stipulated in the Operator's Section of this manual.

CONTROL CARDS

Three general types of control cards are required by the System Load Program: CREAT cards, that specify the system programs wanted; CONFG cards, that specify the standard operation configuration; and Definition cards, that define sections of the System Tape file. The first two card types are supplied by the user; the last type is inserted in the System Tape at the required places.

CREAT Cards

CREAT cards are used to specify which of the "optional" system programs of the System Tape are to be included in the POS being built. An "optional" system program (see Figure 3) must be specified in the operand field of a CREAT card to have it included in the user's system. If the operand field does not specify the inclusion of an optional program, then this program will be omitted.

System Control Program

In order to include the System Control Program in the user's system, all CREAT cards must have the operand "1301."

SYSTEM PROGRAM	OPERAND
COBOL Processor	COBOL
FORTRAN Processor	FORTRAN
RPG Processor	RPG
Print Program	PRINT
System Control Program	1301

LIBRARY SECTION	OPERAND
COBOL Instructions and Subroutines	COBOLM
IOCS Instructions and Subroutines	IOCS
Users Instructions and Subroutines	USER, c or USER, cu The first operand is used if the source statements are on cards. "c" is the number of the channel to which the card reader is connected. The second operand is used if the source statements are on tape. "cu" represents the channel and unit number, respectively, of the tape unit from which the source data will be read.

Figure 3. Optional System Elements and Their CREAT Card Operands

Optional System Programs

Figure 3 lists the optional system programs and indicates the operands that must be placed in the operand field of a CREAT card. (To make the list complete for ready reference, the operands required for the System Control Program are included in Figure 3.) The use of the CREAT card and the other control cards necessary for the inclusion during a Build Run of user macro-instructions in the POS on disk storage is outlined in the "Library" section of this manual under "Inclusion of User Routines at POS Creation."

All other programs on the System Tape are automatically included in every POS built by the user.

The operands selected can be written in any order in the operand field of the CREAT card with a comma placed between each entry. More than one CREAT card can be used.

CREAT Card Examples

Figures 4 through 6 show some examples of this control card.

The sample CREAT card coding shown in Figure 4 specifies that the POS to be built contains the COBOL processor, the FORTRAN processor, the Print Program, the library section of COBOL subroutines, user subroutines to be read in from Tape Unit Number 5 on Channel 1, and the System Control Program. Note that the Autocoder Processor is not optional; it is automatically included.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		CREAT	FORTRAN, 13Ø1
0 2			

Figure 5

The card shown in Figure 5 specifies that the POS to be built contains the FORTRAN Processor and the System Control Program.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		CREAT	RPG, 13Ø1, USER, 1
0 2			

Figure 6

This sample CREAT card (Figure 6) requests the RPG Processor, specifies that user supplied subroutines are to be read into the Library from a card reader on channel 1, and requests the System Control Program.

CONFG Cards

CONFG cards define the standard operating configuration. One card from each of the types shown in Figure 7 must be included in the input packet for the Build Run or a Tape Update Run.

NOTE: The user can change the standard operating configuration during a processor or maintenance and auxiliary run by including the appropriate CONFG cards in the control packet for these runs. The only exception is the CONFG card that specifies SYSAREA1 (this card specifies the cylinders on which the POS is stored and may be used only on a Build or Tape Update Run).

Line 3 5	Label 6 15	Operation 16 20	OPERAND 21 25 30 35 40 45 50 55 60 65 70
0 1		CREAT	COBOL, FORTRAN, PRINT, COBOLM, USER, 15, IOCS, 13Ø1
0 2			

Figure 4

INITINPT

INPTELEM

PRINTELEM

PUNCHELEM

HDRCHECK

SUPPRESS

MACHSIZE

OBJSIZE

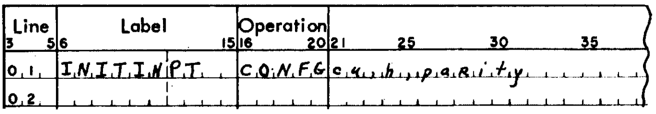
SYSAREA1
SYSAREA2
SYSAREA3
SYSAREA4

RELOCATE

Figure 7. CONFG Card Types

Initial Input Unit

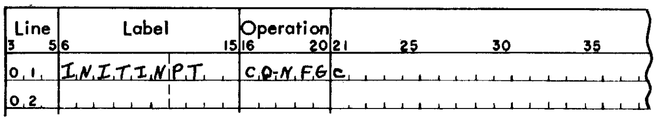
This CONFG card specifies the input device from which CONFG cards (if any) are to be read to change the standard operating configuration prior to a processor, maintenance, or auxiliary run. The two formats of this CONFG card are shown in Figures 8 and 9.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	INITINPT	CONFG	cu,h,parity
02			

Figure 8

The format in Figure 8 specifies that the CONFG cards are on tape. The "c" represents the channel number of the tape unit, the "u" represents a digit specifying the unit number, the "h" represents a digit that specifies whether or not the tape has a header label ("0" if there is no label, "1" if there is a label); the "parity" entry can be either "ODD" or "EVEN."



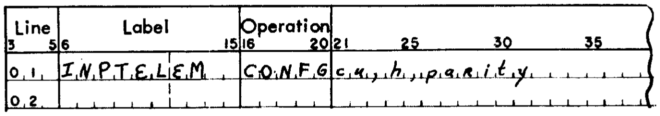
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	INITINPT	CONFG	c
02			

Figure 9

This format (Figure 9) specifies that the CONFG card data is on cards. The "c" represents the channel to which the card reader is attached.

Source Input Unit

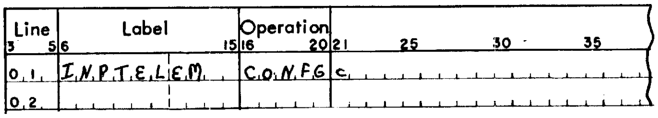
This card specifies the input device from which the source program will be read, The three formats of this card are illustrated in Figures 10 through 12.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	INPTELEM	CONFG	cu,h,parity
02			

Figure 10

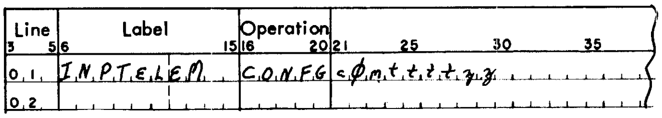
This format (Figure 10) specifies that the source language program is in card-image form on tape. The operand intries are the same as those shown for Figure 8.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	INPTELEM	CONFG	c
02			

Figure 11

This format (Figure 11) specifies that the source language program is on cards to be read in from the card reader on channel "c."



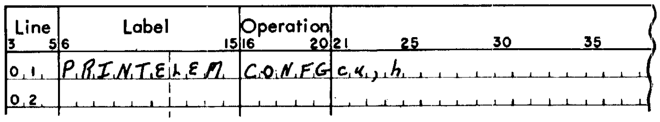
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	INPTELEM	CONFG	cØmttttzz
02			

Figure 12

This format (Figure 12) specifies that the source language program is in disk storage: "c" represents the channel number, "0" is the access mechanism number, "m" represents the module, "tttt" represents the beginning track number, and "ZZ" represents Home Address 2.

Print Unit

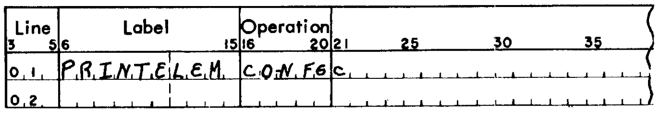
This card specifies the device on which the object program listing is to be placed. The three possible formats for this are shown in Figures 13 through 15.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	PRINTELEM	CONFG	cu,h
02			

Figure 13

This format (Figure 13) specifies that the listing is to be on tape. The operand entries are the same as shown for Figure 8, except for parity.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	PRINTELEM	CONFG	c
02			

Figure 14

This format (Figure 14) specifies that the listing is to be on the printer attached to channel "c."

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	PRINTELEM	CONFG	cØmttttzz
0 2			

Figure 15

This format (Figure 15) specifies that the listing is to remain in disk storage. The operand entries are the same as shown for Figure 12.

Punch Unit

This card specifies the device on which the object program is to be placed. The three formats are shown in Figures 16, 17, 18.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	PUNCHELEM	CONFG	cu,h,r
0 2			

Figure 16

The format shown in Figure 16 specifies that the object program is to be placed on tape. The variable operand entries "c," "u," "h," and "parity" are the same as shown for Figure 8, except for parity. "r" represents a digit that specifies whether or not the tape is to be rewound after the object program is placed on it: "1" specifies that the tape is to be rewound after each object program is placed on it; any other character (including blank) specifies that the tape is not to be rewound.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	PUNCHELEM	CONFG	c
0 2			

Figure 17

This format (Figure 17) specifies that the object program is to be punched into cards using the punch attached to channel "c."

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	PUNCHELEM	CONFG	cØmttttzz
0 2			

Figure 18

This format (Figure 18) specifies that the object program is to remain in disk storage. The operand entries are as discussed for Figure 12.

Header Label Checking

The user can have the System Control Program check standard IBM 1410 header labels without tape marks of all input tapes. The format of this card is shown in Figure 19.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	HDRCHECK	CONFG	n
0 2			

Figure 19

In Figure 19, the operand "0" indicates no header label checking, "1" specifies that tape labels are to be checked.

Suppression of Punch or Print Output

The user can have the POS suppress either punch or print output by means of the card shown in Figure 20.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	SUPPRESS	CONFG	n
0 2			

Figure 20

The operand of the card shown in Figure 20 can be "0" for no suppression, "1" to suppress punch output, or "2" to suppress print output.

Processor Machine Size

This card is used to specify the number of core-storage positions of the machine on which the processor run will take place. (See "Processor Runs.")

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	MACHSIZE	CONFG	n
0 2			

Figure 21

The operand in Figure 21 can be "3" for a 40,000-position system, "4" for a 60,000-position system, or "5" for an 80,000-position system.

Object Machine Size

This card specifies the number of core-storage locations of the system on which the object program will be run.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	OBJSIZE	CONFG	n
0 2			

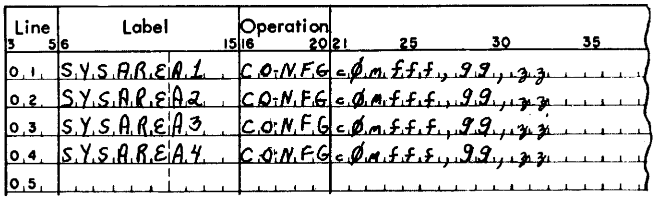
Figure 22

The operand entry in Figure 22 can be "1" to specify a system with 10,000 positions, "2" for 20,000 positions, "3" for 40,000 positions, "4" for 60,000 positions, or "5" for 80,000 positions.

POS System Area

These four cards define the system areas required by POS. The SYSAREA1 card specifies the cylinders on which the POS built by the user is to be stored. The other three cards specify working storage (see Figure 1). The formats for these cards are shown in Figure 23.

NOTE: All four of these cards are required even though all System Area cylinders are contiguous on one module.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	SYSAREA1	CONFG	cØmfff,gg,zz
02	SYSAREA2	CONFG	cØmfff,gg,zz
03	SYSAREA3	CONFG	cØmfff,gg,zz
04	SYSAREA4	CONFG	cØmfff,gg,zz
05			

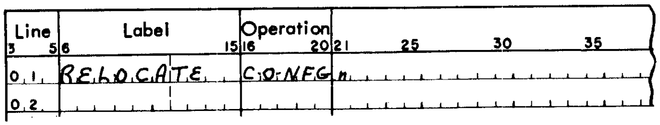
Figure 23

The operand entries (Figure 23) are "c" for the channel number, "0" for the access mechanism, "m" for the module number, "fff" for the beginning cylinder number, "gg" for the number of cylinders (01-99) in this area, and "ZZ" for HA2.

NOTE: The operand entry for HA2 can contain more than two characters. If so, only the first two characters will be used by POS.

Relocation

This card specifies whether or not the object program is to be relocatable.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	RELOCATE	CONFG	n
02			

Figure 24

The operand of the card shown in Figure 24 is "0" if the object program is not to be relocatable, or "1" if the object program is to be relocatable.

Standard Operating Configuration on System Tape Supplied

The System Tape, as supplied by IBM, contains parameters that define a "standard operating configuration," except for SYSAREA1 through SYSAREA4 assignment which must always be supplied by the user. Figure 25 shows that operating configuration on the System Tape as supplied. The user may create this configuration through the use of corresponding CONFG cards, or may use other CONFG card parameters as desired; but in either case a complete set of CONFG cards is required.

OPERATING CONFIGURATION	STANDARD AS SUPPLIED
Initial Input Unit	Card Reader, Channel 1
Source Input Unit	Card Reader, Channel 1
Print Unit	Printer, Channel 1
Punch Unit	Punch, Channel 1
Header Label Checking	No Checking
Suppression of Punch and/or Print Output	No Suppression
Processor Machine Size	40,000 positions
Object Machine Size	10,000 positions
POS System Areas	Not Supplied
Relocation	No Relocation

Figure 25

THE BUILD RUN DECK

The user arranges the control cards as shown in Figure 26.

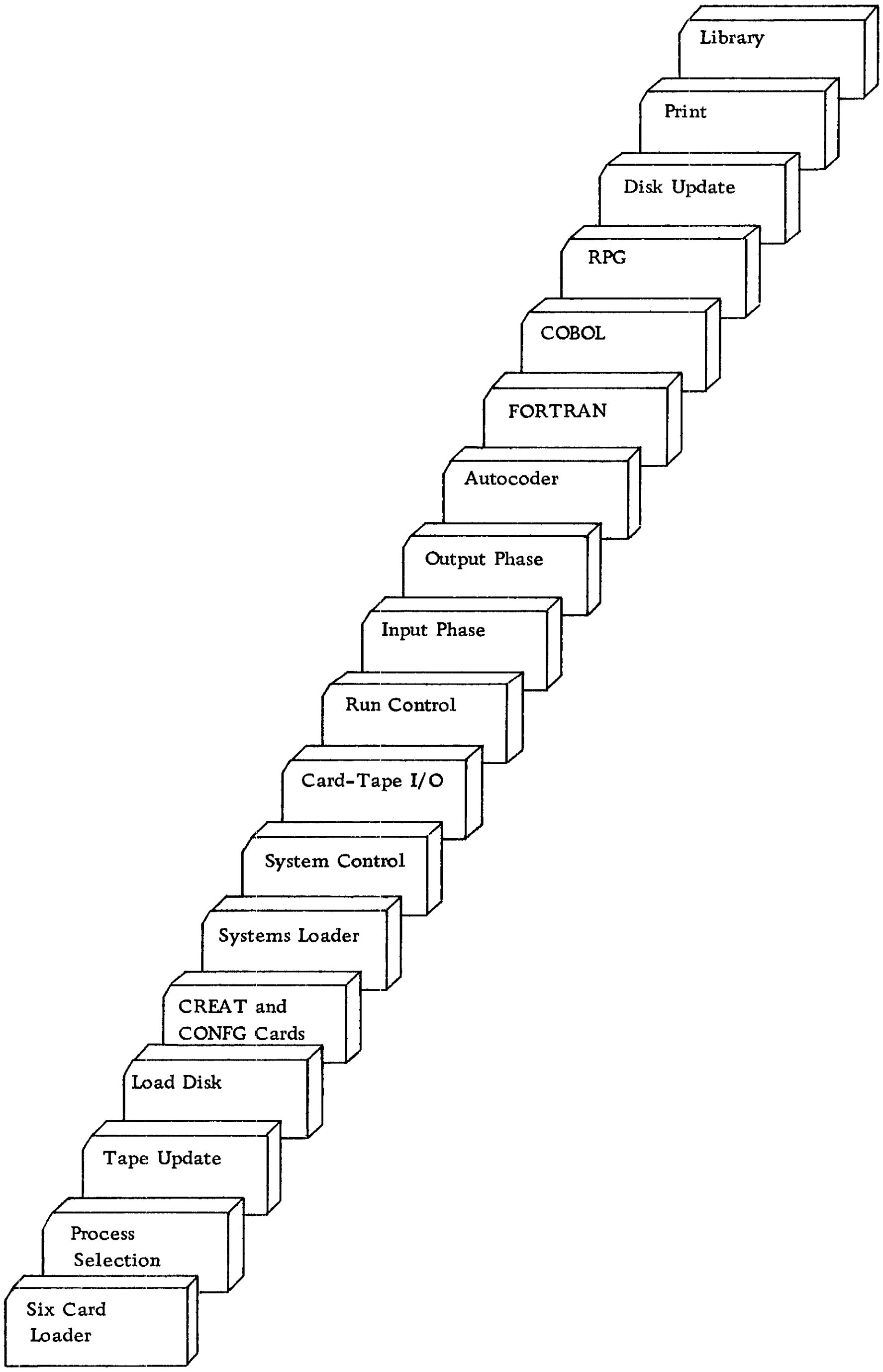


Figure 26. Build Run Deck

GENERAL

During a processor run, source language statements are translated into the required machine-language object program. The POS controls the selection of the required processor, performs input and output functions for the processor, and performs error checking and correction.

The user supplies the POS with control cards that identify the processor required (the RUN card) and indicate the end of each source program (the EOJ card). These cards are described below.

CONFG cards (see "The Build Run") can be included with each source program if the user desires to temporarily modify the standard operating configuration.

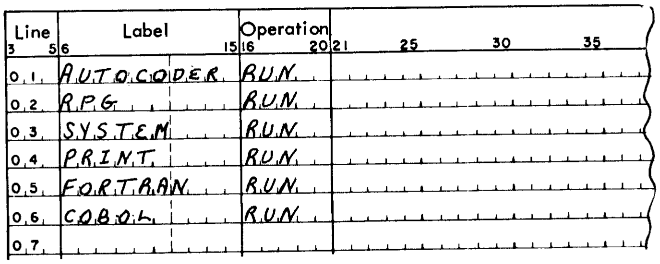
The user can, when using the POS, batch his jobs together so that many source language programs are translated during a single machine run. If jobs are batched, the proper processor will be selected for each job, CONFG cards can be included with each source program to modify the standard operating configuration.

CONTROL CARDS

The control cards used for a processor run are described below.

RUN Card

The RUN card signifies to the System Control Program the type of run requested. It is placed before each source program.



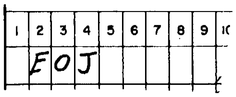
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	AUTOCODER	RUN	
0 2	RPG	RUN	
0 3	SYSTEM	RUN	
0 4	PRINT	RUN	
0 5	FORTRAN	RUN	
0 6	COBOL	RUN	
0 7			

Figure 27

Figure 27 shows the allowable formats of the RUN card. (The Print card is included in Figure 27 so that the user has a complete table of all RUN cards that may be required under the POS.) Operands, if any, for each of the above cards are specified in the discussions on each of the individual processors included in this section.

EOJ Card

The last card of each source program should be followed by an EOJ card (Figure 28).



1	2	3	4	5	6	7	8	9	10
	E	O	J						

Figure 28

(If the EOJ card is omitted, the System Control Program will process all cards up to the next RUN card or until an end-of-file condition occurs.)

AUTOCODER CONSIDERATIONS

General

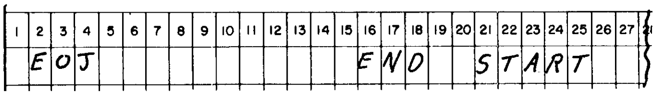
It is assumed, in the material that follows, that the programmer is thoroughly familiar with the material contained in the IBM 1410 Data Processing System Bulletin "Autocoder: Preliminary Specifications," Form J24-1433.

The Autocoder Processor treats all cards between the AUTOCODER RUN card and the END card as a single source program. (The EOJ card is used by the POS.)

It is recommended that the first card of the source program deck (immediately following the RUN card) be a JOB card. A JOB card inserted at this place will cause an identifying header line to be printed on each page of the assembly listing. If the RUN card is not followed by a JOB card, then the processor will use a dummy JOB card. Additional JOB cards inserted into the source deck will perform the functions described in the Autocoder bulletin.

END Cards

The END card is necessary to indicate the end of each source program and to provide the initial entry point for each program (a branch to the first instruction to be executed after the resulting object program has been loaded during an object run). The EOJ card, required by the Input Program of the POS, can be combined with the END card as shown in Figure 29.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	E	O	J												E	N	D			S	T	A	R	T		

Figure 29

In Figure 29, START is a label representing the entry point of a program.

The source deck will also be terminated by an EOJ card, a RUN card for the next program, or an end-of-file condition. That is, the END card may be omitted and the program will process properly with one of the above indicating the end of the source program. Under these conditions, a dummy END card is created by the processor, but an initial entry point will not be provided.

CTL Cards

CTL cards will be treated as Comment cards. Control functions under the POS are performed by CONFG cards.

LOAD Cards

A LOAD card may be inserted anywhere within the source program deck. This will result in one of the following conditions depending upon the output options specified in POS for the source program:

1. If the object program is punched in cards, the standard IBM nine-card load program will precede the object-program deck.
2. If the object program is written on magnetic tape, the standard IBM nine-card load program will precede the object program on tape.

Restrictions

The following restrictions apply when the Autocoder Processor is run within the POS:

1. Extra listings are not provided.
2. The ORG ADDR, S statement is not implemented.
3. The Processor has the following maximum table limits.

Machine Size (Core Storage Positions)	40,000	60,000	80,000
Literal Table (Literals)	500	1,800	3,100
Incomplete EQU Table (Incomplete EQU Statements)	120	120	120

Figure 30

The literal table contains all literals plus all incomplete EQU statements. Each incomplete EQU statement occupies the space of three literals. The number of incomplete EQU statements can be minimized by placing the EQU statements in the source program so that they are defined as they are processed.

The incomplete EQU table contains all EQU statements that are:

1. Undefined
2. At a level of indirectness of 2 or higher while being incomplete at the point where they appear in the source deck. For example, in Figure 31, the first statement would fit this criterion and thus would enter the incomplete EQU table.

```
          .
          .
          .
A   EQU   B
B   EQU   C
C   DCW   #5
          .
          .
          .
```

Figure 31

Autocoder Run Deck Setup

Figure 32 shows the Autocoder Deck Setup.

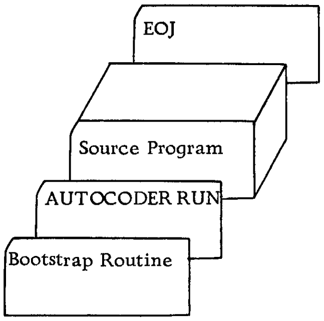


Figure 32. Autocoder Deck Setup

RPG CONSIDERATIONS

General

It is assumed, in the material that follows, that the programmer is thoroughly familiar with the IBM 1410 Data Processing System Bulletins, "Report Program Generator for the IBM 1410: Preliminary Specifications," Form J24-1433, and "Report Program Generator for the IBM 1410--Supplement for 1301 Disk Storage," Form J28-0256.

The RPG source deck must start with an RG control card and must end with an asterisk card. The POS requires that the RPG RUN card precede this deck and that an EOJ card follow.

Four asterisk cards must be inserted at the following places within the source program: between input specification cards and data specification cards; between the data specification cards and the calculation specification cards; between the calculation specification cards and the format specification cards; and following the format specification cards.

If a conversion routine is written in Autocoder language, it must be placed immediately behind the RG control card and it must be followed by a double asterisk card.

NOTE: There is the possibility of a loop with a no-record-found message (40151 NRF -- see list of Halts, Messages and Corrective Actions for POS runs) occurring during a RPG compilation. This can be caused by the asterisk card following the input specification cards not being in its proper place.

The processor searches for a resulting-condition punch in column 43 of the input specification cards, and, if these are not previously terminated by the asterisk card, the processor will often erroneously read the entire deck without encountering another column 43 punch. If the NRF message occurs, the deck should be examined for proper ordering.

Assembly Option

Usually the RPG Run will result in an assembly listing by Autocoder and an object program deck. The user has the option of specifying that the processor compile only the Autocoder source program.

This option is selected by placing an "H" in column 3 of the RG cqntrol card.

Suppression of Punch and/or Print

The options to suppress punch and/or print output as stated in column 4 of the RG control card are no longer valid. These options are controlled, under POS, by CONFG-card entries.

RPG Run Deck Setup

Figure 33 shows the RPG Deck Setup.

COBOL CONSIDERATIONS

General

Specifications for preparing source language programs in the COBOL language are contained in the "IBM COBOL General Information Manual," Form F28-8053, and the 1410 Data Processing Systems Bulletin titled, "COBOL-Additional Specifications," Form J28-0232. It is assumed that the programmer is familiar with the contents of these publications.

To process a COBOL source program, POS requires a RUN card (COBOL RUN) immediately preceding the source program deck and an EOJ card immediately following the source program deck. All statements (cards) between these two POS control cards will be treated by the processor as part of the source program. The omission of the EOJ card causes the Processor System to recognize either an end-of-file condition on the input unit or the RUN card of the following job.

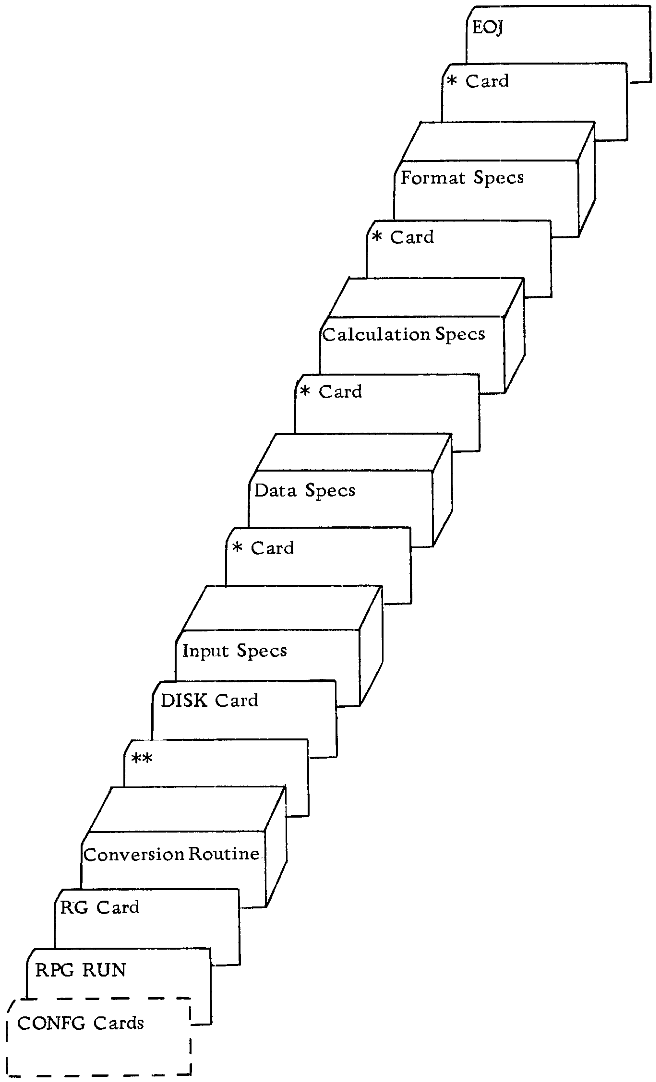


Figure 33. RPG Deck Setup

COBOL Run Options

The COBOL processor run normally results in a full COBOL compilation and Autocoder assembly. Output consists of the COBOL source program listing (including the cross reference dictionary and diagnostic messages), the Autocoder listing, and the object program deck. Placing the operand AUTOCODER in the COBOL RUN card (col 21) has the same effect on the processor run as leaving the operand field blank. The user can alter the processor run by placing one of the following operands in the COBOL run card.

1. DIAGNOSTIC -- This operand will cause termination of the run after diagnosis of source program errors. The output will consist of the COBOL source program listing, cross-reference dictionary, and diagnostic messages.

2. INTERMEDIATE -- This operand will cause a full COBOL compilation to be made, but no Autocoder assembly is produced. Output consists of the COBOL source program listing (including the cross-reference dictionary and diagnostic messages) and the Autocoder source program deck. The Autocoder source deck will be placed on the I/O unit specified by the CONFG card PUNCHELEM. No Autocoder listing is produced.

FORTRAN CONSIDERATIONS

General

Specifications for preparing source programs in the FORTRAN language are contained in the 1410 Data Processing System Bulletin titled, "FORTRAN for the IBM 1410: Preliminary Specifications," Form J24-1468. It is assumed the programmer is familiar with the contents of this publication.

To process source programs in the FORTRAN language, POS requires a RUN card (FORTRAN RUN) immediately preceding the source program deck and an EOJ card immediately following the source program deck. (See Figure 34, FORTRAN Deck Assemblies.)

The FORTRAN processor requires that the first card in the source program deck be a Beginning-of-Program (BOP) card, made up as follows:

Columns					
16-18	21-30	34-35	39-40	49	51-55
BOP	name	fixed	float	opt.	memory size

Column	Description
16-18	BOP (beginning of program)
21-30	name. If a main program, leave blank; otherwise, fill in name of the sub-program. Name must be left justified, starting in column 21.
34-35	fixed integer size, right justified; if blank, a 5 is assumed. The maximum value is 20.
39-40	floating point fraction size, right justified; if blank, an 8 is assumed. The maximum value is 18.
49	Output Options
	blank Output is as specified in the POS configuration cards.
	1 Output will consist of an Autocoder source program listing, standard Autocoder listing, and object program deck.
	2 Output will consist of an Autocoder source program deck, Autocoder listing, and object program deck.
	3 Output will consist of an Autocoder source program dеck and listing, and a standard Autocoder listing and object deck.
	5 Output will consist of an Autocoder source program listing. Run terminates after production of this listing.
	6 Output will consist of an Autocoder source program deck. Run terminates after production of this deck.
	7 Output will consist of an Autocoder source program deck and listing. Run terminates after production of these items.
51-55	memory size of both the compiling and object program machine: assumed 39,999 unless otherwise specified, (must be identical to configuration cards).

Batch Compiling

If it is desired, the user may process several FORTRAN source programs in one run. This is accomplished by placing an EOJ card at the end of each FORTRAN source program in the series, and a FORTRAN RUN card at the beginning of each program in the series. (See Figure 34, Fortran Deck Assemblies.)

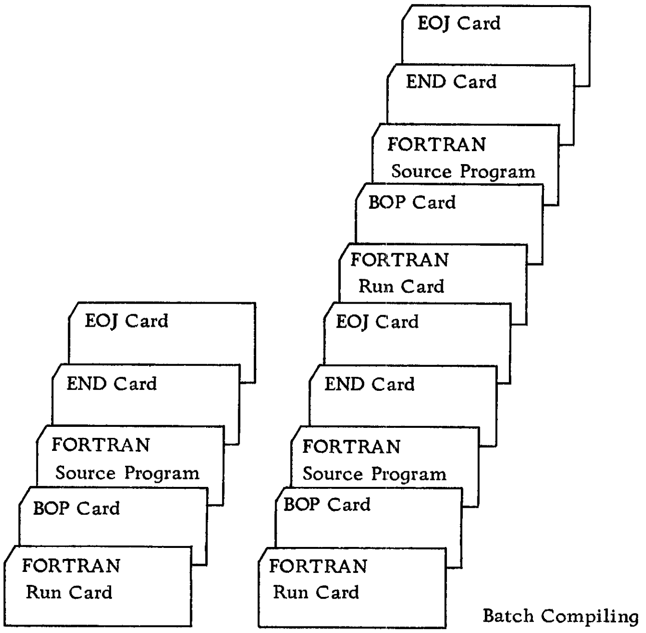


Figure 34. FORTRAN Deck Assemblies

GENERAL

The Update Run performs the maintenance function of updating (changing, deleting, and adding to) the POS. The Print Run performs the Auxiliary function of printing out selected sections of the Library.

THE UPDATE RUN

The Update Run is used to maintain the System. Tnere are two different types of Update Runs available with the POS:

-Tape Update Run

-Disk Update Run

The Tape Update Run is described in this section of the manual; and the Disk Update is detailed under the heading DISK UPDATE PROGRAM. Whenever an Update Run is performed the cards specifying the desired changes should be arranged in a sequence corresponding to that shown in Figure 26. The tape Update Run is used to alter or duplicate the System Tape. Under the direction of control cards supplied by the user, the Tape Update Program can be used to delete, add, or modify a phase of a System Program, an entire System Program, or sections of the Library. The user can also place onto the System Tape the parameters that will determine his Standard Operating Configuration.

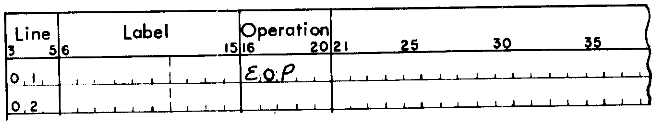
Definition Cards

Definition cards are control cards that indicate to the Update and System Load programs, the beginning and end of various phases of system programs, and the Library. (A phase is a logical block of a system program or library sub-section that is handled as a unit. See "Updating the Library" in this section.) The System Tape, as supplied by IBM, has definition cards inserted at the required places.

A description of these cards and their functions are given below. The user must insert the appropriate cards when altering the System Tape by means of an Update Run.

EOP Card (End-of-Phase Card)

This card must follow each phase of a system program. It indicates, during the Build Run, that a complete phase has been formed within core storage and that the System Load Program can proceed to transfer this phase to disk storage. The format of this card is shown in Figure 35.

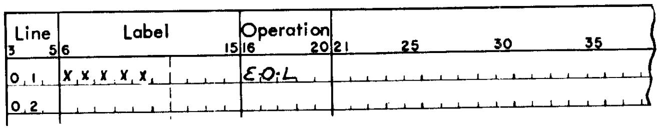


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		EOP	
0 2			

Figure 35

EOL Card (End-of-Library Section Card)

An EOL card must follow each section of the Library. This card indicates through the System Load Program that a complete section of the Library has been processed. The format of this card is shown in Figure 36.



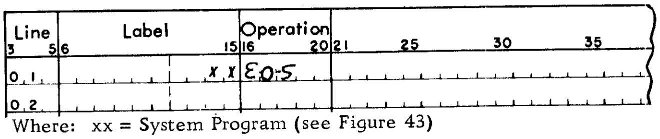
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	xxxxx	EOL	
0 2			

Where: xxxxx = name of section

Figure 36

EOS Card (End-of-System Program Card)

An EOS card must follow the last EOP card of a system program. This card indicates to the System Load Program that a complete system program has been loaded. The format of this card is shown in Figure 37.



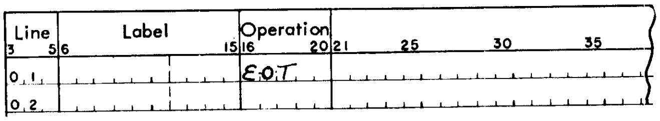
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	xx	EOS	
0 2			

Where: xx = System Program (see Figure 43)

Figure 37

EOT Card (End-of-Tape Card)

An EOT card must follow the EOL card in the USER section of the library (see "Updating the Library"). The format of this card is shown in Figure 38.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		EOT	
0 2			

Figure 38

NOTE: The USER section is the last file on the System Tape.

START Card

The START card must be the first card of each system program and each Library section. The format of this card is shown in Figure 39.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	xxxxxxxxxx	START	yyyy
02			

Where: xxxxxxxxxx = label as shown in Figure 40
yyyy is "TAPE" if a system program uses card/tape IOCS
yyyy is blank otherwise

Figure 39

SYSTEM PROGRAM OR LIBRARY SECTION	START CARD LABEL
System Control Program	SYSCONTROL
Run Control Program	RUNCTRL
Input Program	INPUTSYS
Output Program	OUTPUTSYS
Print Program	PRINTPROG
Autocoder Processor	AUTOCODER
COBOL Processor	COBOL
FORTRAN Processor	FORTRAN
RPG Processor	RPG
IOCS Library Section	ZIOCS
COBOL Library Section	COBOLM
User Library Section	USER

Figure 40. START Card Labels

Header Cards

The first two cards of each phase (except phases LDISK, UDTAP, MONTR) must be header cards, as shown in Figures 41 and 42. These cards specify the name of the phase, the upper and lower boundaries, and the address of the first instruction. The first header card has the format shown in Figure 41.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
01	=ØØØØØ=ØØØ	ØØ=xx	xxx
02			

Where: xxxxx = name of the phase (NAMEX)

Figure 41

The second header card has the format shown in Figure 42.

Line 3 5	Label 6 15	Operation 16 20	OPERAND 21 25 30 35 40
01	=ØØØØ5=ØØØ	ØØ=ee	eee=aaaaa=uuuuu=lllll
02			

Where: eeeee = address of the first executable instruction
aaaaa = address of origin of the phase
uuuuu = update address. This is 19999-- (lllll -- aaaaa)
lllll = last position used by this phase (= or < 39997)

Figure 42

Updating a Phase

In the material that follows "NAMEX" is used to represent the unique name of the phase as specified on the first header card. See Figure 41. Following is a list of the phase names.

Contents of 1410/1301 POS systems tape:

6-CD. LDR
MONTR
UDTAP
LDISK
LOADX
SYSCL
CDTIO
RNCTL
INPUT
OUTPT
DIOCS
MAGEN
MAG2A
FAZ3A
FAZ3C
FAZ4A
FTSUP
PHF30
PHF35
PHF38
PHF40
PHF42
PHF45
PHF50
PHF80
PHF90
COBOL
PH 05
PH 07
PH 10
PH 15
PH 16
PH 20
PH 21

PH 22
PH 23
PH 24
PH 25
PH 26
PH 40
PH 50
PH 51
PH 54
PH 60
PH 61
RPG1
RPG2
RPG3
RPG4
RPG5
RPG6
RPG8
RPG9
RPG10
RPG11
RPG7
RPGIN
RPGDA
RPGCL
RPGFM
RPGED
RPG13
EDITO
PRINT
IOCS LIBRARY
COBOL LIBRARY
USER LIBRARY (If Any)

Control Cards

The control cards used to alter the phases on the System Tape are described in Figures 43 through 48.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	NAMEX	UPDAT	NEW
0 2			

Figure 43

All cards following the card shown in Figure 43, up to another update control card or an END card, will be added as a new phase in front of phase NAMEX. The new phase must have the proper definition cards, i.e., the two header cards and an EOP card. If this new phase is to be added at the end of a system program, NAMEX should be punched "xxEOS" where "xx" is shown in Figure 44.

SYSTEM PROGRAM	xx
Autocoder Processor	AU
COBOL Processor	CO
FORTRAN Processor	FO
RPG Processor	RP
System Control Program	SC
Run Control Program	RU
Print Program	PR
Input Program	IN
Output Program	OU

Figure 44

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	NAMEY	UPDAT	HEADER
0 2			

Figure 45

The card shown in Figure 45 will replace the two-card header of phase NAMEX with the two-card header that follows this control card.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	NAMEX	UPDAT	ALL
0 2			

Figure 46

All cards following the card shown in Figure 46, up to another control card or an END card will replace the entire phase NAMEX. The new phase must contain an identifying two-card header.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	NAMEX	UPDAT	PATCH
0 2			

Figure 47

All cards following the card shown in Figure 47, up to another control card or END card, will be inserted after the last data card of phase NAMEX (i.e., before the EOP card that defines the end of phase NAMEX).

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	NAMEX	UPDAT	DELET
0 2			

Figure 48

This card shown in Figure 48 will delete phase NAMEX from the System Tape. The deletion will include the two-card header and the EOP card.

Updating a Program

Control Cards

The control cards used to add or delete a system program to or from the System Tape are described in Figures 49-50.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	SYSNAME	UPDAT	SYSDELET
0 2			

Figure 49

The card shown in Figure 49 will delete the system program specified by SYSNAME from the System Tape. The program will be deleted in its entirety from the START card up to and including the EOS card. The labels represented by SYSNAME are those shown in Figure 50.

SYSTEM PROGRAM	LABEL
System Load Program	LOADER
System Control Program	SYSCONTROL
Run Control Program	RUNCTRL
Input Prograin	INPUTSYS
Output Program	OUTPUSYS
Autocoder Processor	AUTOCODER
COBOL Processor	COBOL
FORTRAN Processor	FORTRAN
RPG Processor	RPG
Print Program	PRINT

Figure 50

To add a system program to the System Tape, use the card shown in Figure 51.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	SYSNAME	UPDAT	SYSTEMADD
0 2			

Figure 51

All cards following this card, up to another control card or an END card, will be added to the System Tape in front of the system program identified by SYSNAME. The deck for the added system must have the START card, header card for each phase, EOP card, and an EOS card as the last card.

Listing the Library from the System Tape

The control cards used to list all macro headers or selected macros and subroutines from the System Tape are described below.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	HEADER	UPDAT	LIST
0 2			

Figure 52

This card causes all macro and subroutine headers, including all Library START and EOL cards, on the 1403 Printer.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	MACRO	UPDAT	LIST
0 2			

Figure 53

This card causes selected macros or subroutines to be listed on the 1403 Printer. Routines are specified in column 16, one per card, in a deck following this card. The pages from the listing are numbered with a Header at the top of each page.

Changing the Standard Operating Configuration

The control card shown in Figure 54 directs the Update program to change the standard operating configuration parameters on the System Tape.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		UPDAT	CONFG
0 2			

Figure 54

The data on all cards following this card up to another control card or an END card, will replace the parameters that are stored on the System Tape that determines the standard operating configuration. The first card of each package must be a CREAT card.

NOTE: As in a BUILD run, a card must be included in the package for each parameter desired on the updated system tape.

Updating the Library

The Library is divided into three sections: IOCS, COBOL, and USER. Each of these three sections is further subdivided into three subsections identified as subsections M, S, and Z. The M subsection is reserved for macro-instructions; the S subsection for subroutines, and the Z subsection for IOCS routines.

All additions to the Library must be in card-image form in the Library format (see Figure 55).

Control Cards

The following control cards are used to update the Library. On these cards NAMEX identifies the phase, x is the subsection (M, S or Z), and SECTN is the section (ZIOCS, COBOL, or USER).

Line 3 5	Label 6 15	Operation 16 20	OPERAND 21 25	75 80
0 1	NAMEX	INSER	X	SECTN
0 2				

Figure 56

IBM INTERNATIONAL BUSINESS MACHINES CORPORATION

FORM X24-6568-0
Printed in U.S.A.

IBM 1410 DATA PROCESSING SYSTEM

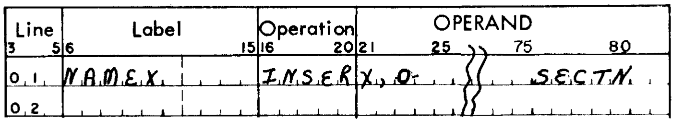
LIBRARY CODING FORM

DATE________ PROGRAM________ PROGRAMMED BY________

Page and Line	L	Label	Operation	Operand and Comments	Identification
1 2 3 4 5	6	7 8 9 10 11 12 13 14 15 16 17 18 19 20 21	22 23 24 25 26	27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74	75 76 77 78 79 80

Figure 55. Library Coding Form

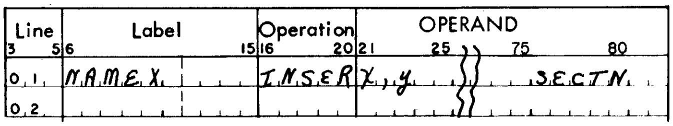
All library format cards following the card shown in Figure 56, up to another control card or an END card, will be inserted at the end of SECTN, subsection x. If NAMEX refers to a macro already in the library, the old macro will be deleted, and the new macro will be put in its place.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	INSER	X,0 SECTN
02			

Figure 57

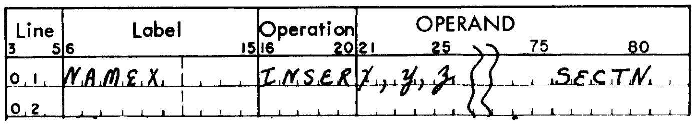
The card following the card in Figure 57 will be inserted as the first statement of NAMEX in subsection x.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	INSER	X,Y SECTN
02			

Figure 58

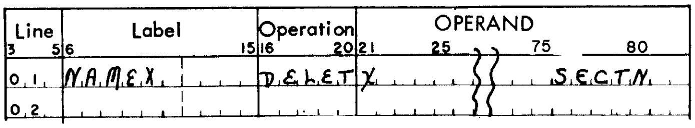
The cards following the card in Figure 58 will be inserted as new statements following statement y of NAMEX in subsection x.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	INSER	X,Y,Z SECTN
02			

Figure 59

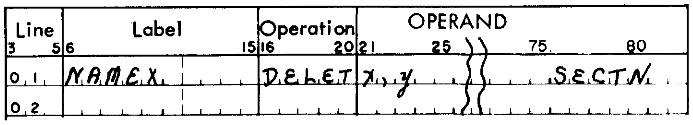
This card (Figure 59) will cause current statements y through z to be deleted. It will also cause the library statements following this control card, up to another control card or an END card, to be inserted, starting with statement y of NAMEX in subsection x.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	DELET	X SECTN
02			

Figure 60

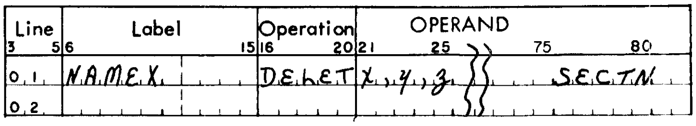
This card (Figure 60) will delete from the System Tape the statements that make up NAMEX in subsection x.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	DELET	X,Y SECTN
02			

Figure 61

This card (Figure 61) causes statement y to be deleted from NAMEX of subsection x.



Line	Label	Operation	OPERAND
3 5	6 15	16 20	21 25 75 80
01	NAMEX	DELET	X,Y,Z SECTN
02			

Figure 62

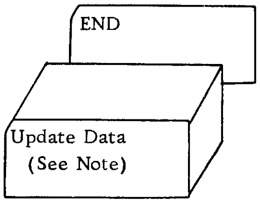
This card (Figure 62) causes statements y through z to be deleted from NAMEX of subsection x.

Duplicating System Tape

To have the Update Program duplicate the System Tape, the user enters a control symbol at the console typewriter during the initiation of the Update Run. See "Update Run," in the Operator's Guide section.

Card Deck Set for the Update Run

Figure 63 shows the card deck setup for the Update Run.



Note:
Update Data consists of:
1. Control Cards (UPDAT, INSER, DELET)
2. Correction (Patches, Header Cards, CONFG Cards, Library Statements)

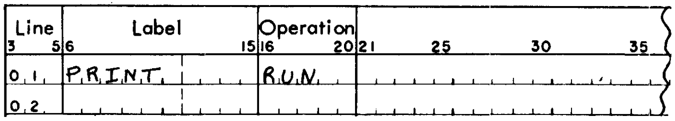
Figure 63. Card Deck Setup -- Update Run

1301 PRINT RUN

The Print Run is used to print out selected sections of the Library from disk storage. The print output is on the device specified by the standard operating configuration or by CONFG cards included in the control package for the Print Run.

Control Card

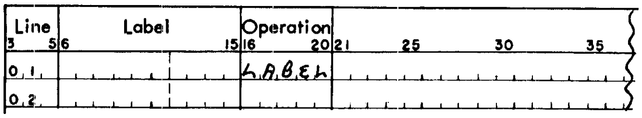
To indicate to the system control program that the Print Run is desired, the control card shown in Figure 64 is used.



Line	Label	Operation	
3 5	6 15	16 20	21 25 30 35
01	PRINT	RUN	
02			

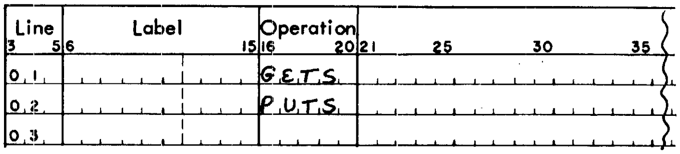
Figure 64

Cards with formats shown in Figure 65 are used to specify the desired library statements. For example, if the user desires to have the subroutines called by the GETS and/or PUTS macro-instructions printed out, he punches cards with GETS and PUTS. See Figure 66.



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,		LABEL	
0,2,			

Figure 65



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,		GETS,	
0,2,		PUTS,	
0,3,			

Figure 66

Card Deck Setup for the Print Run

Figure 67 shows the card deck setup for the Print Run.

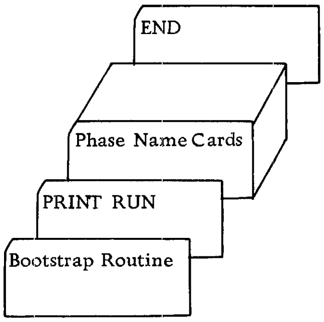


Figure 67. Card Deck Setup -- Print Run

DISK UPDATE PROGRAM

General

The Disk Update program provides the means for maintenance of the POS while on disk. It allows the correction and replacement of system phases, macros and subroutines, communication table information, and will list phase and/or macro directory contents on the console typewriter. It also allows movement of the system from one area of the disk to another.

If any replaced or patched phase is larger than the one it replaces, the system will be moved back the number of tracks required to accommodate the new phase. If there is not enough room between the system-end and library-start for expansion, the library will be moved back one cylinder.

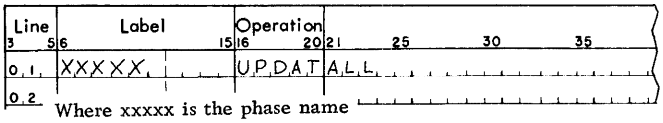
In any track movement required, the directories involved will be updated.

If any updated macro or subroutine requires more area than it originally occupied, the entire updated macro or subroutine will be moved to the end of the Library.

The control cards acceptable to the Disk Update Program are described below.

Control Cards

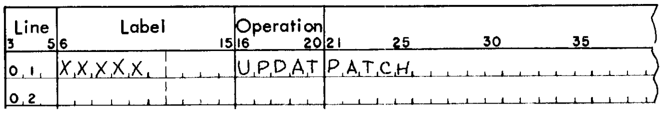
Updating a Phase



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,	XXXXX,	UPDAT	ALL,
0,2	Where xxxxx is the phase name		

Figure 68

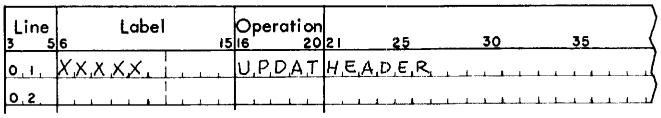
The cards following this card, up to another control card or an END card, will completely replace phase xxxxx on the disk. The appropriate two-card header must precede the phase (see Figures 41 and 42).



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,	XXXXX,	UPDAT	PATCH,
0,2			

Figure 69

The cards following this card up to another control card or an END card, will replace current phase xxxxx on the disk. If a patch falls outside its ORG or END locations, the header must also be updated.

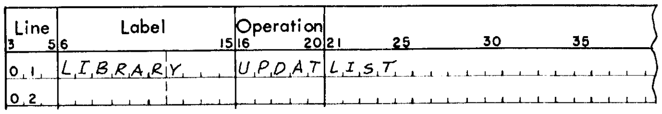


Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,	XXXXX,	UPDAT	HEADER,
0,2			

Figure 70

The two cards following this card will be used to update the ORG, END and ENTRY information on the first track of phase xxxxx.

Listing the Directories



Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0,1,	LIBRARY,	UPDAT	LIST,
0,2			

Figure 71

The Library Directory will be listed on the console typewriter.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	PHASE	UPDAT	LIST
0 2			

Figure 72

The Phase Directory will be listed on the console typewriter.

Moving the System

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		UPDAT	SYSMOVE
0 2			

Figure 73

That portion of System Area 1, which is used for System programs, will be moved to an area specified in a SYSAREA 1 CONFG card following this UPDAT card (Figure 73). If "BOOTSTRAP" is punched in column 6, a new area BOOTSTRAP card will be produced on the PUNCHELEM unit specified in the Communications Table.

Changing the Standard Operating Configuration

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		UPDAT	CONFG
0 2			

Figure 74

The CONFG cards following this card will be used to alter the Communications Table.

Updating the Library

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	XXXXX	INSER	W
0 2	Where W = M for flexible subroutines S for inflexible subroutines Z for IOCS Library		

Figure 75

A new macro or subroutine will be added to the Library, with name xxxxx, into section W.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	XXXXX	INSER	W, X, Y
0 2			

Figure 76

Statement numbers X and Y will be deleted from macro xxxxx, and all new statements following this card will be inserted, beginning with statement number X. If X = Y, only one statement will be deleted before beginning statement insertions.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	XXXXX	INSER	W, X
0 2			

Figure 77

New statements following this card will be added beginning with statement number X + 1. No deletions will be made.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	XXXXX	DELET	W
0 2			

Figure 78

Macro or subroutine xxxxx will be deleted from the disk.

Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1	XXXXX	DELET	W, X, Y
0 2			

Figure 79

Statements x and y will be deleted from macro xxxxx. X may equal Y.

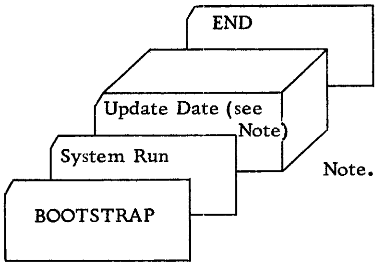
Line 3 5	Label 6 15	Operation 16 20	21 25 30 35
0 1		END	
0 2			

Figure 80

This card causes a return to System Control.

Disk Update Program Deck Setup

Figure 81 shows the Disk Update Program Deck Setup.



Note. - Update Data Consists of:
1. Control Cards (UPDAT, INSER, DELET)
2. Corrections (Patches, Header Cards, CONFG Cards, Library Statements.

Figure 81. Disk Update Setup

Data for punch and print output can be intermixed on one reel of magnetic tape for subsequent off-line printing on an IBM 1401 Data Processing System. This tape can contain assembly listings only, or a mixture of assembly listings and data for the condensed object-program decks. If either or both of these options are chosen, the information on the tape is automatically preceded by the 1401 Print/Punch program (a self-loading program that selectively prints and punches).

What This Program Will Do

The 1401 Print/Punch program has the following features:

1. It is self loading on a 1401 or on a 1410 operating in 1401 mode.
2. It prints assembly listings.
3. It punches condensed card decks from tapes containing intermixed information.
4. It writes the tape records for condensed object program decks, produced by the Fortran processor, as a file on tape for the Fortran relocatable loader.
5. It reconstructs and punches 1410 symbolic card decks from the assembly listings that are on tape. (EJECT cards are not produced when the program reconstructs the symbolic card deck.)
6. It can bypass any selected tape file or group of files controlled by sense-switch settings or control cards. (These are described below.)

Mixed tapes of the type serviced by this program will be used by IBM to distribute listings and program decks to the users of the 1410 system.

Installation program libraries can be conveniently stored on tapes containing intermixed information.

Machine Requirements

The following are the minimum 1401 machine requirements for the program:

1400 positions of core storage
1 IBM 1402 Card Read-Punch
1 IBM 1403 Printer
1 IBM 729 or 7330 Magnetic Tape Unit

NOTE: The system must be equipped with the no-charge RPQ #898148 to properly punch cards. This RPQ modifies the 1401 system to read an 8-2 card column as an A bit and to punch an A bit as an 8-2 (rather than as a zero).

Control-Card and Sense-Switch Options

Figure A-1 shows control-card entries, sense-switch settings, and the options controlled by each. If the sense switches are used, control cards are not required, and conversely, if control cards are used, sense switches are not required. One control card is used to indicate all the options chosen.

NOTE: Do not use control cards and sense switches simultaneously.

Control Card	Sense Switch	Function
L in column 1 blank in column 1	B, normal position B, ON position	Prints assembly listing Bypass printing
P in column 2 blank in column 2	C, normal position C, ON position	Punch condensed decks Bypass punching
blank in column 3 P in column 3	D, normal position D, ON position	Bypass punching Punch symbolic deck
blank in column 4 T in column 4	E, normal position E, ON position	Bypass tape write Write out on tape 4

Figure A-1

Operator's Guide

Using the 1401

1. Mount and ready the tape on unit 1.
2. Mount and ready a scratch tape on unit 4 if a Fortran tape is to be written.
3. Ready the card read-punch.
4. Press CHECK RESET and then START RESET.
5. Press TAPE LOAD.
6. The Print/Punch program will be loaded into core storage and the program will halt. The "Storage Address" lights on the console will indicate "376."
7. Set the sense switches or place a control card in the reader. There will be a machine halt (with a "Storage Address" indication of "376") before each tape file is processed. Thus, the user may change the options chosen for each file. To bypass a file, simply choose all four bypass options.
8. If sense switches are being used, press START. If control cards are being used, press START RESET and then START.

9. After the last file is processed , "EOJ" will be printed on the 1403 printer, and the tape on unit 1 will be rewound.

Using the 1410 in 1401 Mode

1. Place the tape on channel 1, unit 1.
2. Place a scratch tape on channel 1, unit 4 if a Fortran tape is to be written.
3. With the 1410 in 1410 mode:
   a. Clear storage
   b. Display "19000"
   c. ALTER in "$\overset{v}{\text{L}}$%U1001R$\overset{v}{\text{B}}$001$\overset{v}{\text{.}}$"
4. SWITCH the 1410 to the 1401 mode.
5. Branch to 19000 and press START.
6. Proceed as in step 6 in this section under "Using the 1401."

Error Correction

Programmed error-correction procedures are for tape errors only. There will be no programmed halts in case of tape errors.

If a tape-read error occurs, a maximum of nine rereads are attempted. If the read is still unsuccessful, the first 80 characters of the record are punched, and the punched card is selected to the "NP" pocket. The printer will be triple spaced, and "X" will be placed in the first print position, and the entire record will be printed.

If a tape-write error occurs, the program will skip the tape forward and attempt to write the tape. No halts will occur.

NOTE: Normally the 1401 Print/Punch Program will precede the first output listing file on any tape unit specified by the CONFG card PRINTELEM. However, during a Processor Run and after the first job has been processed, it is possible to suppress the 1401 Print/Punch Program as follows:

1. A change of PRINTELEM from 1403 Printer to a tape unit will suppress the 1401 Print/Punch Program from the magnetic tape.
2. A change of PRINTELEM from one tape unit to another will suppress the Program from the second tape.

Thus, by taking advantage of these two conditions, the 1401 Print/Punch Program can be suppressed as the first file on the output listing tape.

The information in this appendix is not required by the user for any POS runs but is to be used during debugging analysis.

Described in this appendix are three tables that are used by the POS for communication between system programs and between the phases of a system program. These tables are: Communication Table, Program and Phase Directory, and Library Directory.

THE COMMUNICATION TABLE

The Communication Table contains the control information (Figure B-1) that is in effect at any one time in the system. Many of the entries in this table reflect the options specified in CONFG-card entries. (These are indicated in Figure B-1 by the appropriate CONFG-card label.)

The Communication Table occupies core-storage positions 00500 through 00999 during any type of run other than the Build Run or Tape Update Run.

The location, description and format of each of the Communication Table entries are shown in Figure B-1.

PROGRAM AND PHASE DIRECTORY

The Program and Phase Directory is recorded in the load mode on a single disk track. The address of this track is given in the Communication Table in core-storage positions 00598-00601.

The program entries are given first on this track, followed by the phase entries.

Program Entries

Each program entry is eight characters in length with the format shown below:

```
v v   v
xxrrrryy
```

Where:

xx indicates the name of the system program (see figure 43).

rrrr is the relative address of the location of the first phase entry for this system program. This address is relative to the first character position on the track.

yy is the number of phases in this system program. (If a system program uses the card/tape IOCS, then an "A" bit will be associated with the low-order "y.")

Phase Entries

Each phase entry is ten characters in length with the following format:

```
v
ttttnnnnnn
```

Where:

tttt is the track address of the first track of the phase.

nnnnnn is the name of the phase. (If the name of a phase is less than six characters, it is left-justified in positions nnnnnn with the unused positions filled with blanks.)

The last entry of the Program and Phase Directory consists of five numeric characters that specify the position of the last character of the last phase entry relative to the first character on the track.

THE LIBRARY DIRECTORY

The Library Directory is recorded in the load mode on a single disk track. The address of this track is given in the Communication Table in core-storage positions 00602-00605.

The entries in the Library Directory consist of the names and locations of all routines in the POS Library.

The first entry in the Library Directory consists of four numeric characters that specify the position of the last character in the Library Directory relative to the first character.

The individual Library Directory entries have the following general format:

```
v
ttttsnnnnn
```

Where:

tttt is the number of the disk track on which the library routine begins.

s represents the subsection to which the library routine belongs. It can be:

M for macros
S for subroutines
Z for IOCS routines

nnnnn is the identifying name of the library routine.

Core Storage Location	Description	Format
00500-00504	Initial Input Unit (INITINPT)	bxxxz
00505-00509	Source Input Unit (INPTELEM)	bxxxz
00510-00514	Print Unit (PRINTELEM) (See Note)	bxxxz
00515-00519	Punch Unit (PUNCHELEM) (See Note)	bxxxz
00520-00528	System Area 1 (SYSAREA1)	camttttthh
00529-00532	Total number of tracks in System Area 1	mmmm
00533-00536	Number of unused tracks in System Area 1 available to the phase being run	mmmm
00537-00545	System Area 2 (SYSAREA2)	camttttthh
00546-00549	Total number of tracks in System Area 2	mmmm
00550-00553	Number of unused tracks in System Area 2 available to phase being run	mmmm
00554-00562	System Area 3 (SYSAREA3)	camttttthh
00563-00566	Total number of tracks in System Area 3	mmmm
00567-00570	Number of unused tracks in System Area 3 available to phase being run	mmmm
00571-00579	System Area 4 (SYSAREA4)	camttttthh
00580-00583	Total number of tracks in System Area 4	mmmm
00584-00587	Number of unused tracks in System Area 4 available to the phase being run	mmmm
00588-00591	Number of tracks in System Area 1 in use	mmmm
00592-00597	Version and Modification Levels of this POS	ppvvll
00598-00601	Track address of the Program and Phase Directory	tttt
00602-00605	Track address of the Library Directory	tttt
00606-00610	Unused	
00611-00619	Address of source input data if this data is in disk storage	camttttthh
00620	Tape Header Label Check (HDRCHECK)	d
00621	Processor Machine Size (MACHSIZE)	e
00622	Directory Indicator -- ON, if Library Directory is in core storage	"B" (on)
00623	First Job Switch (ON, if first job of a batch or if a job changes the input/output configuration)	d
00624	Suppression of Punch or Print (SUPPRESS)	w
00625	Rewind option for Punch output. If the object program punch output is placed on tape, the switch is ON if the tape is to be rewound after each object program is transferred.	d
00626	Loader option. If the object program output is in cards or on tape, this switch is ON if the program output is to be preceded with the IBM standard load program.	d
00627	PST Indicator ON if the Autocoder processor is to supply the symbol table	d
00628	Object Machine Size (OBJSIZE)	o
00629	Relocation (RELOCATE)	d
00630	EOF Indicator ON if the source program input has reached end of file	d
00631	Assembly run switch ON if assembly is to follow compilation. This switch is set by the RUN card operand.	d
00632	Internal POS code. (This is the same as in location 00631 except that this switch is set by the Processor.)	--
00633	Alternate Suppress Switch ON if the particular program being run has suppression option. (The entry in "00624" is the Standard Operating Configuration.)	w
00634	First Card Indicator ON if a phase of a system program has read in a card of a source program which was unanticipated. (This would be the result of an incorrect source program deck setup.)	d
00635-00639	Not used	
00640-00654	Instructions by which system programs pass control from one program to another	--
00655-00714	Communication Words. These are used to indicate the location of files of data. There are five sets of file-words in this field each symbolically, FILST (start of the file) and FILND (end of the file).	
	FILST	cmttttthh
	FILND	tttt
00715-00716	Unused	
00717-00721	Running counter of the highest position assigned by Autocoder	mmmmm
00722	Signal from Autocoder Phase 3 to Autocoder Phase 4	d
00723-00727	Location of the start of the Autocoder Symbol Table	mmmmm
00728-00732	Location of the end of the Autocoder Symbol Table	mmmmm
00733-00747	Autocoder title card parameters	
00748	Recursion Switch for the Autocoder Symbol Table	d
00749-00753	Not used	
00754-00758	Address of the last symbol in the Autocoder Symbol Table	mmmmm
00759-00766	Not Used	
00767-00846	First Card Save Area (this field contains a card image record of the card which caused the switch at 00634 to be set on)	
00847-00851	Name of the phase currently being run	
00852-00998	Unused	
00999	Group Mark with a Word Mark	

Key to Format

a	Access mechanism number. This is always a "0".
b	Input/Output Unit;
	2 Tape
	3 Card Reader
	4 Card Punch
	5 Printer
	6 1301 Disk Storage
c	Channel Symbol:
	@, Channel 1
	*, Channel 2
d	0--No or OFF
	1--Yes or ON
e	Processing Machine Size (Internal Format)
	3 40,000 positions
	5 60,000 positions
	7 80,000 positions
hh	HA2
ll	Modification Level Number
m	Module Number, 0 → 9.
n	Digit from 0 → 9.
o	Object Machine Size (Internal Format)
	0 10,000 positions
	1 20,000 positions
	3 40,000 positions
	5 60,000 positions
	7 80,000 positions
pp	Processor System Number
tttt	Track Number (HA1)
vv	Version Number
w	0 No Suppression
	1 Suppress Punch
	2 Suppress Print
xxx	X-Control Field
z	Branch if any I/O Channel Status Indicator ON
	R -- Channel 1
	X -- Channel 2

Note: If this unit is a tape unit, the data will be in odd parity.

Figure B-1. The Communication Table

INTRODUCTION TO RELOCATION

A relocatable or relocation loader is a program designed to relocate sub-programs and routines in order to place them sequentially into storage, while modifying addresses to reflect changes in locations. Such a loader is needed for incorporating into a single program, at object-time, parts of a program that have been written, assembled, and perhaps debugged independently, and for including standard routines from a library. These sub-programs would have been written and assembled as though each started at the same fixed position in storage, usually location zero. The relocation is accomplished by assigning progressively higher addresses to successive instructions, constants, work areas, etc., within a routine and from one routine to the next.

Obviously those operands which address relocated items must be suitably modified while others must be left unchanged. Examples of the former are the addresses of factors in arithmetic operations and operands of branch instructions; examples of the latter are addresses designating input/output devices and system registers. Most addresses are affected by relocation increments; however, those addresses referencing COMMON may be adjusted either upward or downward.

The loader will either modify a given address or not depending on relocation indicators placed in the load card produced by an assembler. Thus an assembler, in addition to converting symbolic codes and labels to machine language, will analyze each address as to the need for modification under relocation. Its output, in the form of load cards with relocation indications would then become the input to the relocation loader.

An additional function of a relocatable loader is to provide the linkage necessary to pass control from one sub-program to another. This presents no difficulty if the called routine has already been loaded at the time it is called since a simple branch instruction can be placed in the calling routine. However, if the called routine has not yet been loaded and therefore the address of its entry point is unknown, a dummy branch instruction must be put into the calling routine and an indication stored to return at the proper time to fill in the required address. Thus, the use of a suitable assembly routine and a relocatable loader eliminates the need for reassembly when subprograms and routines are to be incorporated into a single program in storage.

FORTRAN RELOCATABLE-PROGRAM LOADER

The FORTRAN Relocatable Loader for the IBM 1410 is designed to load, relocate, and link subprograms and routines written in FORTRAN, processed by the FORTRAN compiler and assembled by the 1410 AUTOCODER processor run under the "RELOCATABLE" option. (See "Autocoder Relocation.") In addition, it makes provision for options to pre-set program switches, to punch the relocated program into cards preceded or not by a load routine or to write it on tape, to reassign tape units to be used by the object program, to obtain the object program from different input devices during loading, and to print on the console typewriter pertinent information about each routine loaded. It will also print a warning message if the loaded program exceeds memory and if called routines have not been loaded.

Storage addresses are modified as described in the section, "Introduction to Relocation." It should be noted that COMMON addresses are assigned at assembly time on the basis of the defined object machine and normally remain unchanged when loaded. However, the designation of either a different object machine with the BOP card; or of a high order address other than the highest with an upper load limit control card, will cause the Loader to appropriately alter the COMMON addresses. It should be noted that there may be duplication in the COMMON address if two or more programs or routines which reference COMMON, and which have been assembled separately, are loaded together.

The Relocatable-Program Loader will operate on the following minimum machine configuration:

20,000 positions of core storage[1] and
1 IBM 1402 Card Read-Punch, Model 2 or
1 IBM 729 Magnetic Tape Unit, Model II, IV, V, VI or
1 IBM 7330 Magnetic Tape Unit

If the programs being loaded cannot be included in core with the Relocatable Loader, an additional Magnetic Tape Unit will be required for overflow.

AUTOCODER RELOCATION

Although the FORTRAN Relocatable Loader is primarily designed for loading programs written in FORTRAN, it can be used for programs written in

[1] The Loader as supplied is coded to occupy storage positions 27000 to 38200. However, it may be placed into any desired part of memory by re-assembling it with the Autocoder processor and a new ORG card.

Autocoder. The FORTRAN compiler translates the FORTRAN statements into Autocoder language; the Autocoder processor converts this program to machine language in condensed load card format. At the same time it analyzes all addresses and punches relocation indicator codes to define the adjustment that must be made to the addresses under relocation. It uses the following pseudo-operations produced by the FORTRAN compiler:

1. TITLE
2. DCWS
3. DCWF
4. DAV

Therefore, a user may write a program or subprogram directly in Autocoder language and obtain a relocatable condensed deck if he specifies "RELOCATABLE" as the operand of the AUTOCODER RUN card and if he includes in the input any of the pseudo-operations named above that are required by his program.

The TITLE Card

The function of the TITLE pseudo-operation is to cause the Autocoder processor to develop information regarding a given routine to be used by the FORTRAN loader in placing the routine in storage and establishing proper linkage to it from other programs. The format of the input card and that of the Autocoder output card are:

```
Column  6      21               31    36    41           72
Input   TITLE  NAMEX------
Output         TITLE NAMEX--- AAAAABBBBBCCCCC...         5...
```

where
NAMEX -- is the name of the routine or is left blank if it is part of the mainline program.
AAAAA -- is the address of the first word of the routine as assembled by Autocoder.
BBBBB -- is the address of the entry point to the routine, obtained from the END card.
CCCCC -- is the number of COMMON positions reserved for the routine.
5 -- is a card relocation code explained later.

The DCWS Card

The function of the DCWS pseudo-operation is to cause the Autocoder processor to develop a transfer to a called routine.

```
Column  1                             16     21
Input                                 DCWS   NAMEX-----
Output  Wxxxxx  W00007WJNAMEX-----
        S       S        S

Column  71   72   73 - 80
Input
Output  S    6    IDENT
```

where
NAMEX -- is the name of the routine called.
xxxxx -- is the load address for the transfer instruction as assigned by Autocoder.
S6 -- is the relocation code explained later.
IDENT -- is the card number and program identification.

The FORTRAN Loader will replace NAMEX ----- by the address of the relocated entry to the routine.

The DCWF Card

The function of the DCWF card is to direct the loader to obtain from the library a routine to be used as an argument of a called routine and to create an adcon for its entry point. It is designed to accomplish the function of the FORTRAN-F card. Thus the combination of a DCWS card and one or more DCWF and DCW cards produce the same result as a CALL SUBROUTINE statement in FORTRAN. For example, the FORTRAN statement

```
         CALL     BOB(SIN,S)
    F    SIN
```

can be accomplished in Autocoder by

```
         DCWS     BOB
         DCWF     SIN
         DCW      S
```

The formats of the input card to and the output card from the Autocoder processor are

```
Column  1                   16      21            71  72  73 - 80
Input                       DCWF    NAME------
Output  WxxxxxW00005WNAME------                   S   7   IDENT
        S      S     S
```

where
NAME -- is the name of a library routine.
xxxxx -- is the address of the location to which the adcon will be loaded.
S7 -- is the relocation code.
IDENT -- is the card number and program identification.

The DAV Card

The function of this pseudo-operation is the same as that of the DA except that it defines an area in the COMMON storage area. The COMMON is used to hold parameters shared by subprograms and routines and is usually in upper memory. The DAV card must be preceded by an ORG card indicating the actual lowest address of the defined area. It is followed by any field definition cards as with the DA, and must also be followed by an RSV card for each label of the DAV statement referenced elsewhere in the program.

```
Column 1          15         21
Input    NAME     ORG        COMMONCTR - K
Input    NAME     DAV        BxL
Input    FIELDA              M.N

Input    RSV      RSV        NAME
Input             RSV        FIELDA
```

and the output is a set of cards of the format:

```
Column 1               15
Output  EWD00201 xxxxx4WA0025300211 WS0021300258 WJ00281
         S              S               S            S
                                      67  72  73 - 80
        VWJ00201 ₿W 0000 LW 0000 B    8   IDENT
         S        S        S
```

COMMONCTR has been equated to the uppermost core storage location. K is the total number of COMMON storage locations assigned to and including this point in the program.

NAME, B, L, FIELDA, M, and N have the same meanings as they do in the DA pseudo-operation.

8 is the card relocation code.

IDENT is the card number and program identification.

The set of output cards consists of execute cards each of which is a one card routine for setting the required word marks for each field in the defined area.

RELOCATION INDICATORS

The Autocoder processor will produce condensed load cards with relocation indicators starting in column 72 and continuing toward column 1. The indicator in column 72 designates to the Relocatable Loader the type of card and is one of the following:

Code	Meaning
0	Card whose contents are not relocatable
1	Standard relocatable card for instructions
2	The final execute card
5	Title card
6	Transfer entry card (from DCWS)
7	F-function card (from DCWF)
8	Execute card - but not the final one - (e.g. set word mark for DA or DAV)
9	End of loading library routine

Beginning in column 71, indicators are punched for each load word contained in the card. The indicator punched in column 71 refers to the first load word in the card beginning in column 13, the one in column 70 refers to the second load word in the card, etc., (these indicators are printed on the Autocoder listing just before the character count for each item). The format of a 1-type card is

```
Column 1     7         13                       69  70  71  72
       WxxxxxWOOOOYYWI...WI...WI......  R   R   R   I
             S         S   S    S

Column 73-75 76-80
       ZZZ   IDENT
```

where

xxxxx -- is the load address of the first instruction on the card.

YY -- is the count of characters to be loaded.

I -- is an instruction of from one to twelve characters.

R -- is the instruction relocation indicator.

ZZZ -- is the card number.

IDENT -- is the program identification.

The indicator for a given instruction shows the presence or absence of an A-field or I address, B-field address, or X-control field, gives the direction of relocation for each address and indicates the presence or absence of an op-modifier. In the following table are shown the indicator codes for possible combinations of address and relocation, the first number of each pair being used if there is no op-modifier and the second if there is.

	B Address			
A-field or I Address:	No Relocation	Upward Relocation	Downward Relocation	No B Address
No relocation	0 or ?	1 or A	2 or B	
Upward relocation	3 or C	4 or D	5 or E	@ or)
Downward relocation	6 or F	7 or G	8 or H	: or (
No A or I address				9 or I
X-Control Field	L	J	K	M

The other indicator codes and their meanings are:

Code	Meaning
N	Adcon -- relocate upward
P	Adcon -- relocate downward
Q	Adcon -- do not relocate
S	Symbolic DCWS card
V	Constant

CONTROL CARDS

Flexibility in the use ot the FORTRAN relocatable-program loader is provided through control cards. These cards are normally included just before the program to be loaded, except as otherwise noted in the explanations; however, they may be included at any time before the option they designate becomes effective. An option in a control card can be reset or changed by including the same type of control card with the new option. The control cards must be punched with the name of the option starting in Column 1 (only the first five columns are used by the loader), the option left-justified in column 30-36 and a C in column 72. The following table is a summary of the options and the assumed values if the control card for an option is not included. An explanation of each type of card follows the table.

Columns	30-36	Assumed Values
LIBRARY TAPE UNIT	n	None
ENTER FROM	nuz	CARDS
FINAL INPUT TAPE	nuz	CARDS
LOWER LOAD LIMIT	xxxxx	00700
UPPER LOAD LIMIT	xxxxx	39999
INTERMEDIATE TAPE UNIT	n	4
REPLACE PRINTER BY	mn	On-line printer
OBJECT TAPE ASSIGNMENT	mn=rrb̸p	(See explanation)
MODULE AVAILABLE 1301	CDDDEEE	1 249 001
CHANNEL NUMBER 1301	(1-4)	@
HOME ADDRESS 1301	(00-99)	00
MEMORY MAP	u	No
WRITE RELOCATED TAPE	u	0
PUNCH RELOCATED DECK	0, 1 or 2	2
CLEAR MEMORY FROM	xxxxx	39999
SENSE SWITCH SETTINGS	uuuuuu	000000

where
u = 0 or 1
xxxxx -- is a core address.
n -- is a Channel 1 tape unit number.
z -- is the number of files.
m -- is a channel number
rr -- is the FORTRAN tape number 01-20.
p -- is the parity.
CDDDEEE -- is module number, first cylinder in area, and number of consecutive cylinders.

LIBRARY TAPE UNIT

Function: To cause the loading of the "Initial package" from a designated tape unit. The "initial package" is a set of routines the user has selected to be included in every program and has placed at the beginning of the Library Tape followed by a record with 9 in position 72 (card relocation indicator). The routines in the "initial package" can be altered by recreating the Library Tape. The use of this option does not preclude the calling of other library routines. This card can occur at anytime in the loading and will cause the Library Tape to be rewound and the "initial package" loaded.

Note 1: Since each call for a routine not yet loaded causes an entry in the forward reference table (maximum 150 entries) the library routines should be loaded before the main program.

Note 2: There are advantages in tape handling in putting the FORTRAN loader at the beginning of the Library Tape.

Option: Column 30 — Number of tape unit (must be on channel 1)
Standard: It is assumed that there is no "initial package."

ENTER FROM

Function: To designate a card reader or a tape unit as the device for reading programs or data to be loaded. An ENTER FROM control card or its image on tape would be placed at the end of the items to be loaded from one device in order to start the loading from another.

Option for Tape (must be in odd parity):
Column 30 — Number of tape unit (must be on channel 1)
Column 31 — 0 if 80-character record form.
1 if mixed record length. (In this case, only 80-character records will be loaded, the others ignored.)
Column.32 — 1 - 9 to indicate the number of consecutive routines (files) to be loaded from this tape.
Option for Cards:
Columns 30-35 — CARDS
Standard: CARDS

FINAL INPUT TAPE

Function: To cause the loader to read, from a specified tape unit, the routines that have been called but not loaded when the end-of-program blank card is reached. At that time the loader examines the forward reference table for the existence of such routines and, if any, prints a message, halts, and, when restarted, reads from some input device to find the missing routines.

Option:	Column 30	Number of tape unit (must be on channel 1)
	Column 31	0 if 80-character record form. (In this case, only 80-character records will be loaded, the others ignored.) 1 if mixed record length.
	Column 32	1-9 to indicate the number of consecutive routines (files) to be loaded from this tape.
Standard		CARDS

LOWER LOAD LIMIT

Function: To cause the loader to begin relocating items at a specified core storage address. This control card, which can be included at any place in the program being loaded, causes the loader to reorigin to the specified load point. This could cause overlapping of items already loaded.

Option:	Columns 30-34	Core storage address
Standard:		00700

Note 1: Positions 00505-00655 are required by the loader for storing the object time and load time control tables. They must not be overlaid.

Note 2: The loader will not relocate any address nor any instruction assigned by the Autocoder processor below 00700. Therefore, it is advisable that an Autocoder ORG 700 be used at assembly time.

UPPER LOAD LIMIT

Function: To indicate a maximum relocated address, normally the uppermost core position available on the object machine. This control card should be used when loading programs into computers with more than 40K storage to indicate the additional storage locations.

Option:	Column 30-34	Core storage address (usually highest)
Standard:		39999

INTERMEDIATE TAPE UNIT

Function: To designate a tape unit for use by the loader as a scratch tape on which to place items which would overlay the load routine itself. At the end of the loading, the load routine will be cleared and the items loaded into their proper locations.

Option:	Column 30	Number of tape unit (must be on channel 1)
Standard:		4

Note: If the WRITE RELOCATED TAPE option is used, the tape unit designated here will be used for that purpose and no intermediate items will be written.

REPLACE PRINTER BY

Function: To cause print instructions in the program being loaded to be changed to write a tape record on the designated unit. FORTRAN assumes a 132 position printer on-line for the object program and this option would be necessary for a completely tape-oriented system.

Option:	Columns 30,31	Channel and tape unit
Standard:		On-line printer assumed

OBJECT TAPE ASSIGNMENT

Function: To designate the assignment of specific tape units for the tape numbers used in a FORTRAN statement. An OBJECT TAPE ASSIGNMENT control card should be included for each tape unit number used in FORTRAN statements, unless the standard assignment is acceptable.

Option:	Column 30	Designated channel
	Column 31	Designated tape unit
	Column 32	=
	Columns 33,34	FORTRAN tape number
	Column 36	U for even parity B for odd parity

Standard:

FORTRAN	STANDARD ASSIGNMENT	
Tape Number	Channel	Unit
1	1	1
2	1	2
3	1	3
4	1	4
5	1	5
6	1	6
7	1	7
8	1	8
9	1	9
10	1	0
11	2	1
12	2	2
13	2	3
14	2	4
15	2	5

FORTRAN Tape Number	STANDARD Channel	ASSIGNMENT Unit
16	2	6
17	2	7
18	2	8
19	2	9
20	2	0

MODULE AVAILABLE 1301

Function: To designate an area consisting of one or more consecutive 1301 disk cylinders to be used by the disk I/O library routine called by a FORTRAN program which includes disk statements. Four such areas may be designated by four control cards, provided all are on the same channel, as specified in the CHANNEL NUMBER 1301 control card.

Option:	Column 30	Module number
	Columns 31-33	First cylinder number in the area
	Column 34	Number of consecutive cylinders
Standard:		1 249 001

Note 1: The first control card of this type replaces the assumed value; subsequent cards define additional areas.

Note 2: The values from these control cards are loaded into core locations 00620-00652 where they are used by the disk I/O library routine.

CHANNEL NUMBER 1301

Function: To specify a channel and mode to be used by the disk I/O library routine called by a FORTRAN disk statement.

Option:	Column 30	1 specifies channel 1, nonoverlap mode 2 specifies channel 1, overlap mode 3 specifies channel 2, nonoverlap mode 4 specifies channel 2, overlap mode
Standard:		1, i.e. channel 1, non-overlap mode

HOME ADDRESS 1301

Function: To specify an HA2 which is common to all 1301 tracks in the areas used by the disk I/O library routine.

Option:	Column 30,31	00-99
Standard:		00

MEMORY MAP

Function: To cause the name, entry point and amount of relocation to be printed on the console typewriter for each routine at the time it is loaded. An example is given below.

Option:	Column 30	1 if the memory map is wanted 0 if the memory map is not wanted for succeeding routines.
Standard:		0

Example:

```
R  INTRPRT       00700      00000
R  IOCOMMON      02244      01544
R  OPENING       11572      10872
R  CLOSEIO       11598      10898
R  ENTRYONE      11624      10924
R  ENTRYTWO      11650      10950
R  XXXAOP1       11676      10976
R  XXXAOP2       11785      11085
R  XFIX          11939      11239
R  FLOAT         12170      11470
R                12970      12270
R  SORT          14716      14016
```

The name of the routine is followed by the address of the initial executable instruction as determined by the TITLE card and the relocatable loader. The second number is the increment which must be added to every address on the assembly program listing to find its relocated value. The blank name followed by the above mentioned addresses represents the main program. Any routine(s) named after the main program consist of those which the loader has called into memory.

WRITE RELOCATED TAPE

Function: To cause the loader to write the relocated program on the tape unit normally used for intermediate tape instead of loading into storage. The tape will be self-loading; however, it must be loaded from the same channel and unit on which it was written.

Option:	Column 30	0 Do not write a relocatable tape 1 Write a relocatable tape
Standard:		0

Note 1: This option precludes the necessity for an intermediate tape. However, the tape unit designated (or the assumed one) in the INTERMEDIATE TAPE UNIT will be used by this option.

Note 2: This option precludes execution immediately following the relocation run.

PUNCH RELOCATED DECK

Function: To cause the loader to punch the relocated program into cards in load card format, preceded or not by a five card load routine.

Option:	Column 30	0 Terminate punching 1 Start punching 2 Start punching and precede the deck with a load routine.
Standard:		0

Note: This option does not effect nor is affected by the INTERMEDIATE TAPE UNIT and WRITE RELOCATED TAPE options.

CLEAR MEMORY FROM

Function: To cause the loader to put out a Clear Memory routine and to designate an upper bound for the part of memory to be cleared. This should be used with either the WRITE RELOCATED TAPE or PUNCH RELOCATED DECK options to clear memory prior to reloading and perhaps to preserve a COMMON area.

Option:	Columns 30-34	Core storage address
Standard:		39999

SENSE SWITCH SETTINGS

Function: To set the initial condition of the six simulated sense switches used in FORTRAN. If this control card is used, all six switch settings must contain a valid punch (0 or 1). Just prior to execution of the relocated program, the loader will load the switch settings into core locations 00505-00510. No word marks are written.

Option:	Columns 30-35	0 set corresponding switch off 1 set corresponding switch on
Standard:		000000

PROGRAMMING CONSIDERATIONS

Autocoder

1. All programs utilizing relocation will begin assembling at 00700 unless otherwise specified by ORG card.
2. Addresses below 00700 are not relocatable.
3. A nonrelocatable indicator will be assigned to a label which is equated to an actual address.

```
        LABEL   EQU   15000
```

4. The following will be treated as a relocatable address constant:

```
        DCW   LABEL-NAME
```

5. The following combination is not allowed

```
        LABEL   EQU   NAME
        NAME    EQU   900
```

since the former implies relocatability and the latter nonrelocatability.

Loader

1. It is assumed that the first record will be read from the card reader. If the installation is tape-oriented, the user must modify the LOADER as follows: INPUTF must be changed to a 2 and TAPE must be altered to designate the proper channel, parity, and unit. This can be done by changing the cards in the symbolic deck of the loader and reassembling it.
2. When using the automatic restart procedure, the tape table is not reinitialized; the loader will utilize the tape assignments from the previous job.
3. It is suggested that all subroutines be loaded prior to the main program, since, otherwise, the forward reference table might overflow. There are provisions for only 150 such forward references.
4. Format statements cannot be read at object time.

LOGIC DESCRIPTION

The Loader consists of the following main parts:

A TABLE which lists the meaning of each of the relocation indicators, RIEV: Relocation Indicator Expansion Values.

A TABLE OF THE RELOCATION INDICATORS (ARGUMENT) matched with the address (function) where the meaning of the indicator is expanded, RIEV: This table is called the Relocation Table and referred to by the loader as RTABLE.

CONSTANTS used by the loader.

OBJECT TIME CONTROL TABLE AND LOAD TIME CONTROL TABLES, which are loaded into core at object time. Control cards may be used to modify the values in the tables.

INPUT AND OUTPUT AREAS used by the loader.

TABLE 1 ..... A 750 character table which stores the name and first executable instruction address of each routine that the loader processes. The table will accommodate 50 routine titles and the corresponding addresses.

TABLE 2 ..... A 2250 character table which stores the address and names of Subroutines called by DCWS and DCWF instructions, if the subroutine called has not been loaded prior to the call. The table will handle 150 forward references.

AN INITIALIZER ROUTINE which sets up all initial values in the loader-initializer.

AN INPUT ROUTINE which decides whether input is from cards or tape and loads accordingly.

A DECIDER ROUTINE, which examines column 72 at each card to determine the subroutine that will process the card. Each of the 6 subroutines process their own type of record. There is little programming that is common to any two of them. However, subroutines like OUTPUTRTN and LADDING are used during several of the other routines.

SUBROUTINES: The Title Card Routine: TITLECDRTN, does the following:

1. Stores the address from the cards-ORG, EX, present common.
2. Places the subroutine name and first executable address relocated, in table 1.
3. Searches TABLE 2 to see if the routine has been called during the loading. If so, a branch record output occurs with the address of the relocated called subroutine.

If the routine has not yet been called, the location of the request is noted and a "patch" record will furnish the necessary information when the call is completed.

4. Writes the name, first executable instruction and amount by which the routine is relocated on the console if called for. See Memory Map Control Card.

The Transfer Entry Card Routine: TRANSENTRY, performs the following functions:

1. Relocates load address. Branches to LADDING subroutine.
2. Searches TABLE 1 to see if the subroutine called has been loaded yet.
   a. If so, the subroutine address is placed in a branch instruction, or DCWF, output record.
   b.. If the called routine has not been loaded, the address of the branch instruction and program name are entered in table 2.

Program and execute card routine (P and E): has the responsibility of relocating address on type 0, and 1 cards.

1. P and E Routine - This routine increases or decreases the A or B addresses for relocatable instructions. Index 8 is used to point to the proper relocation indicator. Index 9 refers to the A or B address of each instruction on the card.

The relocation indicator is looked up in RTABLE. The function in RTABLE is the address of a set of values in RIEV table which are used for each instruction.

The values indicate if the instruction has 1 or 2 addresses, the total instruction length, and A and B field relocation characteristics.

The routine loop is repeated for each instruction on the card. The two entry points for this routine are:

PANDE01: entry point for program cards (relocatable)
PANDE02: entry point for execute cards (not relocatable)

Basically this routine relocates one instruction element at a time. X8 and X9 are used to control the flow of the data through the routine.

X9 is used to track through the instructions.

X8 tracks through the relocation indicators.

Thus all instructions which affect an instruction to be loaded have an address INPUT + X9 and all instructions which affect a relocation indicator have an address of INPUT + X8.

A program card has its first element starting in column 13 and its first relocation indicator in column 71. Thus the DECIDERRTN sets 12 into X9 and branches to PANDE01. PANDE01 relocates the Load address. X8 is then set to 70 thus relocation starts with the following situation.

```
Col 1                              Col 71          Col 72
                                                     0
W     W     W                                        1
SLLLLLS000CCS.. .. .. ..            I

Input      Input + X9               Input + X8
                (12)                      (70)
```

A table look up is done on the character at INPUT + X8 in table RTABLE to see where, in RIEV, this indicator is explained.

RTABLE has entries of the form:

v
nnn01

where I is the indicator and nnn is the amount to add to the base address of Table RIEV to get the address of the expansion values of I.

As a result of the table lookup nnn is stored in X6. The entry at RIEV + nnn will be one of three forms:

a. @0@ (meaning no addresses are relocatable)
@LL@ (length of element including W at its beginning) S
b. @1@ (1 address to be relocated)
@LL@ (length of instruction)
@NN@ (number to be added to X9 so that counter will be at units position of address to be relocated)
+DLADD) (relocation increment address)
(or
+DHADD) DLADD = Low Add. (upward relocation)
DHADD = High Add. (downward relocation)
c. @2@ (2 addresses to be relocated)
@LL@ (length of instruction)
@NN@
+DLADD (or +DHADD)
@NN@ Same as (2)
+DLADD (+DHADD Same as (2)

The program then tests C(RIEV + X6) Core (1) to see if it is zero. If so, it does no relocation; but steps X9 up by C(RIEV + 2 + X6) (i.e. LL), subtracts 1 from X8 and starts over again with the next element.

If C(RIEV) + X6), = 1 Core (2) , the program
1 Adds C(RIEV + 4 + X6) to X9 (i.e. NN)
2 Sets WM at INPUT -4 + X9
3 Stores C(RIEV) + 9 + X6) in X5
4 Adds X5 to INPUT + X9 & clear WM at INPUT -4 + X9
5 Subtracts C(RIEV + 4 +X6) back off from X9
6 See if there is a 2nd address to relocate
   (a) If so, repeat above.
   (b) If not, go to next element.

Pictorially: Let AAAAA be address to be relocated

```
          W       W      W
INPUT:  SLLLLLS000CCS0AAAAA....    I
                          Input + X9     Input + X8
RIEV
RIEV + nnn  @1@                X6 = 00nnn
            @13@               X8 = 00070
            @06@               X9 = 00012
             +DLADD
```

Instructions

a. Add RIEV + 4 + X6 to X9
so that X9 sets at 18

```
   W      W      Wv
   SLLLLLS000CCS0AAAAA        Input + X9
```

b. Set WM at INPUT -4 + X9

c. STORE C (RIEV + 9 + X6) into X5

```
      X5 + DLADD               (Delta Low Address)
```

d. Add 0 + X5 to INPUT + X9

```
    WLLLLLW000CCW0AAAAA
    S       S      S          Input + X9
                         +DLADD
```

e. Clear WM at INPUT -4 + X9

f. Subtract RIEV + 4 + X6 from X9 so that the counter returns to its initial value.

g. Compare RIEV + X6 to 2 if equal relocate 2nd address; if not Add RIEV + 2 X6 to X9
Subtract 1 from X8
and

h. Proceed to next element.

2. LADDING ROUTINE..... This routine is a closed subroutine used during the execution of the PANDE or TRANSENTRY routines. The value in the DHADD or DLADD counter is added to the card load address. The related address is compared to the high core position or assigned end of core to determine if the program exceeds core. If so, a console typewriter message is printed and the loading halts. If the loading is within core limits control is returned to the main routine.

The Control Card Routine: CONTROLRTN, processes all control cards.

The Final Execute Card Routine: Indicates the end of the present program. It sees how far the present routine goes in core (exclusive of common) and sets the address counter (IL) to that amount +1 for the next program.

The BLANKCDRTN: Handles the EOF condition and the sensing of a blank card. It will:

1. Test to see if this was the library being loaded; if so it goes to the control card routine to reinitialize I/O for the next input.

2. Test to see if Table 2 is not empty, indicating that subroutines are missing.

3. Punches, or writes on tape, the object time control table if required.

4. Test for relocated tape and autoload; if so it rewinds the tape, moves a load instruction and a clear memory routine to Low core and branches to them.

5. If no auto/load program halts.

Program Assembly Error Conditions

The following conditions, arising during program assembly, may indicate that System Area 1 is inadequate for assembly of the program. Corrective action is to perform a Build Run, providing additional System Area 1 cylinders.

1. Recursions, or successive alphabetic sequences of the symbol table

2. A loop in FAZ4A

3. The printing of object cards in an assembly listing

4. The punching of assembly listing lines in an object deck

5. Inconsistent errors in the operation code and operand fields of an assembly listing

6. Incorrectly computed addresses

7. A disk address developed outside of the allocated systems areas (this problem usually results in a "no record found" error message)

LIBRARY

SUBROUTINES

In addition to those listed in "IBM 1410 FORTRAN," Form J24-1468, the following are also provided:

XXXAOP3	Subscript calculator, triple dimension
XXXAOP2	Subscript calculator, double dimension
XXXAOP1	Subscript calculator, single dimension
ENTRYTWO	List of Entire Arrays for Input and Output
ENTRYONE	List of Simple Variables for Input and Output
OPENING	Opens Files
INTRPRT	Floating Add, Subtract, Multiply and Divide
SINCOS	Series Utilized for Sine and Cosine
CLOSEIO	
ENFILETYP	
IOCOMMON	
FQTOVFL	
EXPONFF	
EXPONII	
REWINDTP	
BACKSPACE	
FDVDCHK	
FACOVFL	
EXPONFI	
SENSELIGHT	

Availability

These routines are contained in a tape which is supplied by IBM. A program listing, resequenced symbolic deck, and a condensed deck may be obtained from this tape by utilizing the method contained in the system control section of Users and Operator's Guide #1410-PR-108.

Special care should be exercised when punching the 1410 FORTRAN II Subroutine Tape, #1410-FO-913 (or listing or punching any 1410 tape), to use an IBM 1410 or an IBM 1401 with the ¢ character no charge RPQ #898148, thus insuring correct interpretation of the ¢ character as a 2-8 punch, and vice versa.

Loader-Library Tape

The FORTRAN Relocatable-Program Loader and relocatable subroutines may be placed on tape to facilitate loading programs and related subroutines.

Relocatable subroutines should be divided into two groups:

1. Programs which should always be included in an object run.
2. Programs which should be included only if called by other routines.

The former group should include the following routines:

IOCOMMON
INTRPRT
OPENING
CLOSEIO
XXXAOP1
ENTRYONE
ENTRYTWO

The loader allows 150 branches to routines which are not loaded when the transfer occurs, the above routines being called most frequently could cause table overflow in large programs.

Library Control Card

A Library Control card is identified by a 9 punch in column 72 and performs two functions:

1. The LIBRARY TAPE loader control card transfers loading to the library tape specified; relocation and loading continue until a Library Control card is encountered.
2. The library search uses the Library Control card to indicate the end of each routine.

Constructing a Library Tape

The Library Tape is comprised of 80-character card images in odd parity. Once the Library Deck is built up, the Library Tape is created with an unblocked card-to-tape operation. A tape mark should end the Library Tape. The following paragraphs describe the building of the Library Deck.

The first item of the deck should be a 5-card loader. If a tape-oriented FORTRAN Loader is desired, the standard 5-card loader should be altered as follows for the Library to operate on drive n, channel 1:

CARD	1	Columns	12-13 and	46-47	from 11 to Bn
CARD	3	Columns	49-50		from 11 to Bn

The second item of the Library Deck should be the FORTRAN Relocatable Loader. After the FORTRAN Loader, there should be a control card, ENTER FROM, specifying an I/O device on which the loading of the program will start.

The fourth item of the Library Deck should be the essential package of library routines, followed by a type 9 card (the LIBRARY CONTROL Card).

The nonessential routines, each one followed by a type 9 LIBRARY CONTROL Card, should follow the essential package.

INCLUSION OF USER ROUTINES AT POS CREATION

User-written macro-instructions may be incorporated into the User Library section of the POS through a Build Run. Input may be punched cards, or card-image records on magnetic tape.

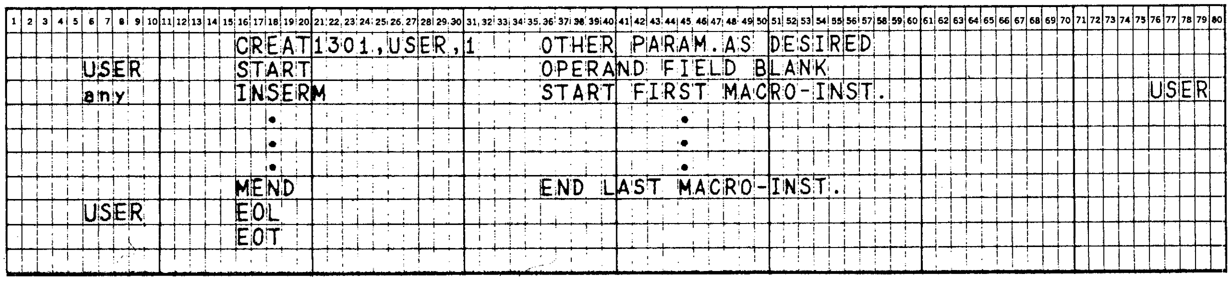
For punched-card input, these steps are required:

1. The CREAT card for the Build Run must include the operands USER, 1 to indicate that input to the User Library section is from the card reader (channel 1).
2. The first INSER card in the card reader must be preceded by a USER START card, with a blank operand field, as shown in Figure 39.
3. The MEND card of the last macro-instruction must be followed by a USER EOL card and an EOT card.

The required coding is outlined in Figure C-1.

For tape input, these steps are required:

1. The CREAT card must include the operands USER, cu. The operand cu is the channel and unit on which the input exists; cu must represent the System Tape if the macro-instructions were placed on the System Tape through a previous tape update run.
2. The input tape must begin with an INSER card-image.
3. The tape must end with a USER EOL card-image followed by an EOT card-image.



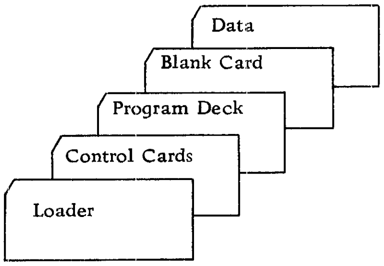
```
               CREAT1301,USER,1    OTHER PARAM.AS DESIRED
     USER      START               OPERAND FIELD BLANK
     any       INSERM              START FIRST MACRO-INST.                 USER
                 .                          .
                 .                          .
                 .                          .
               MEND                END LAST MACRO-INST.
     USER      EOL
               EOT
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

• Figure C-1. Control Cards for Macro-instruction Inclusion at POS Creation

OPERATING INSTRUCTIONS

1. If the Relocatable Loader and Library Routines are on the same tape reel:
   a. Mount Loader-Library on a channel 1 unit. Provide a Library Tape UNIT control card for this unit if any Library Routines are desired.
   b. Mount scratch tape for intermediate tape requirements, if any, on channel 1. Provide INTERMEDIATE TAPE UNIT control card.
   c. Arrange the cards in the reader in the sequence shown below:



   d. At Console enter:
      1. clear memory
      2. display 00001
      3. alter 00001 to $\overset{v}{\mathrm{L}}\%\mathrm{BX00257}\$\overset{v}{\mathrm{J}}\mathrm{00257b}\overset{v}{.}\overset{v}{.}$
         (X=loader-library tape unit)
      4. set mode switch to run status;
      5. depress reset start
2. If CARD LOADER:
   a. Mount scratch tape for intermediate tape on a channel 1 unit and provide an INTERMEDIATE TAPE UNIT control card, if desired.
   b. Mount other required tape and punch appropriate control cards.
   c. Arrange control cards in the reader in the sequence shown in 1a.
   d. At CONSOLE ENTER:
      1. clear memory
      2. display 00001
      3. Alter 00001 to $\overset{v}{\mathrm{L}}\%\mathrm{1100257}\$\overset{v}{\mathrm{J}}\mathrm{00257b}\overset{v}{.}$
      4. set mode switch to run status
      5. depress reset start
3. Relocated Tapes and Decks
   a. Relocated Tapes are self-loading and should be mounted on the unit number (channel 1) on which they were written. Relocated decks should be preceded by the standard five card load routine. (This is one of the options when punching a relocated deck.)
   b. No control cards or memory maps may be used.
   c. Place Data cards in Reader preceded by blank card.
   d. At console enter:
      1. clear memory
      2. display 00001
      3. alter 00001 to $\overset{v}{\mathrm{L}}\%\mathrm{BX00257}\$\overset{v}{\mathrm{J}}\mathrm{00257b}\overset{v}{.}$
      4. set mode switch to start
      5. depress reset start
         X = Unit relocated program was written on.

MESSAGES

BAD CARD TYPE COL 72

Explanation: The card relocation code (col. 72) is not acceptable, i.e. it is not 0,1,3,5,6,7,8,9,C.

Action: Clear reader and check figure punched in column 72 of this card.
Clear memory and reload program, or
Ignore card by pressing start, or
Press computer reset and begin loading another program.

BAD CONTROL CARD

Explanation: Invalid control card or mispunched card. Columns 1-5 checked for type C cards.

Action: Clear reader, check card in control card explanation. Clear memory and reload program.

MISSING SUBRTN XXXXXXX

Explanation: Subroutine called during loading of program has not been found on library tape. Program can not be executed until this routine is loaded. Message 4a or 4b will follow, depending on whether or not FINAL INPUT TAPE option is used.

Action: Note message 4a or 4b.

START TO READ TAPE

Explanation: The loader will attempt to read final input tape in search for missing subroutine indicated in message 3.

Action: Check tape listing to see if Subroutine is included on input tape. If so, press start and loading will resume.

START TO READ CARDS

Explanation: The loader will attempt to read the missing subroutine indicated in message 3 from the card reader.

Action: Place missing subroutine deck in card reader. Press start to resume loading.

ERROR, NOTREADY, NO TRANSFER

Explanation: Loader has entered error routine and displayed condition.

Action: If error condition is NOT READY, ready unit and press start to continue. If error condition is NO TRANSFER, restart job from beginning.

CORE STORAGE EXCEEDED

Explanation: The program, as being loaded, will exceed storage capacity.

Action: Either reload with a higher memory specified or recompile.

THE END

Execution of loading is completed. Load and go was not specified.

ERb (actual I/O instruction)
ERb (actual I/O instruction)
NTb (actual I/O instruction)

Correct condition specified and press start. In tape it attempts to read ten times; pressing start will not rectify situation. Should restart the job with a new tape.

This section contains information required by machine operators for running the POS using 1301 Disk Storage.

THE START INSTRUCTION

All runs are initiated by loading the Start instruction into core storage (location 00000) from the console typewriter. The format of the Start instruction follows:

```
 v         v
zLxxx00012$g
```

z   for the Build Run:

1 if control cards are to be read in from a card reader on channel 1.

2 if control cards are to be read in from a card reader on channel 2.

A if control cards are to be read in from magnetic tape on channel 1, unit 0.

B if control cards are to be read in from magnetic tape on channel 2, unit 0.

blank if control-card parameters are to be read from the System Tape.

for Processor and Print Runs:

blank if the control cards are to be read in from the Initial Input Unit specified by the xxx field of the Start instruction.

any other alphameric character if control cards are to be read in from the Initial Input Unit as specified by control parameters in disk storage.

xxx   The X-control field for the Initial Input Unit. (This must be for the non-overlapped mode.)

g   Represents a "Branch If Any I/O Channel Status Indicator On" operation code for the channel to which the Initial Input Unit is attached. This must be:

R for channel 1, or

X for channel 2.

BUILD RUN (Load Disk)

To Initiate a Build Run

1. Mount the System Tape on any unit on either channel. (The xxx field of the Start instruction must represent the X-control field for this tape unit.)
2. Place the required control cards in the Initial Input Unit.
3. Alter in, starting at location 00000, the Start instruction.
4. Press COMPUTER RESET and then START.
5. At the message "30500 LD-UD," press INQUIRY REQUEST and enter "LD" at the console.
6. Press INQUIRY RELEASE. This causes the POS to be loaded into disk storage.
7. The end-of-job message is "00519 DISK LOAD END OF JOB."

NOTE: If the Build Run is to be performed using cards for input, the first three card blocks illustrated in Figure 26 must be removed, and a standard load program substituted. The user's selection of CREAT and CONFG cards is placed between the load program and the Systems Load program deck.

TAPE UPDATE RUN

To Initiate a Tape Update Run

1. Mount a scratch tape on any available unit attached to either channel.
2. Mount the System Tape to be updated on any available unit attached to either channel.
3. Alter in, starting at location 00000, the Start instruction.

NOTE 1: It is recommended for most efficient operation, that the system tape and the scratch tape be on different channels.

NOTE 2: The 1, A, and B entries of the Start instruction have no effect during a Tape Update Run, but if a Build Run is to follow this Tape Update Run, the appropriate character should be inserted. If the user desires to have a copy of the System Tape created by the Tape Update Program, place the letter "C" in the first position of the Start instruction.

4. Press COMPUTER RESET AND START.
5. At the message "30500 LD-UD," press INQUIRY REQUEST, and enter "UD" at the console.
6. Press INQUIRY RELEASE.
7. At the message "00514 ENTER I/O UNIT ASSIGNMENTS," press INQUIRY REQUEST and enter xxx, yyy at the console. (xxx is the X-control field for the scratch tape. yyy is the X-control field of the unit from which the update data are to be read.)
   Example: If the scratch tape is on unit 5 of channel 2 and the update data are to be read from the card reader, enter the following seven characters: "¤B5,%11."

NOTE: If an UPDATE LIST card is being used, type bbbbyyy. If a System Tape copy is requested, enter xxx,%11 (seven characters).

8. The end-of-job message is "10509 UPDATE COMPLETED."

BOOTSTRAP ROUTINE

The Bootstrap routine must be placed into core storage prior to a Processor Run, Print Run, or Disk Update Run. This routine can be read into core storage from the Initial Input Unit as defined in the Start instruction or it can be entered directly into core storage via the console. If the Bootstrap routine is entered from the Initial Input Unit, the card shown in Figure 82 should be used.

If the user enters the Bootstrap routine via the console, Alter in, starting at location 00012, the equivalent data shown in Figure 83.

NOTE: If the Bootstrap routine is entered via the console, the Start instruction is not used.

PROCESSOR, PRINT, AND DISK UPDATE RUNS.

To Initiate a Processor, Print or Disk Update Run

1. Ready the Initial Input Unit and the Source Input Unit.
2. If the Bootstrap routine is in the Initial Input Unit:
   a. Alter in, starting at location 00000, the Start instruction.
   b. Press COMPUTER RESET.

   If the Bootstrap routine is entered via the console:
   a. Press COMPUTER RESET.
   b. Address set to "00018."
3. Press START.
4. The end-of-job message is "30599 EOB."

COMPOUND RUNS

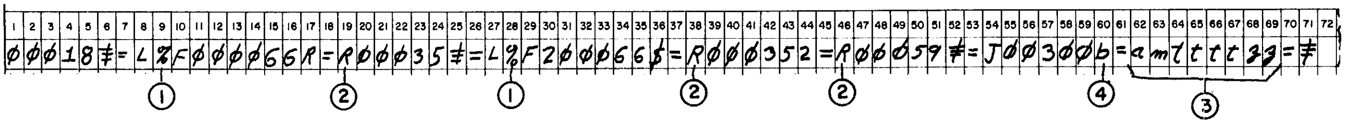
To Load the POS Immediately After a Tape Update Run

1. Press COMPUTER RESET and then START.
2. At message "30500 LD-UD," press INQUIRY REQUEST and then enter LD at the console.
3. Press INQUIRY RELEASE.

To Initiate a Processor Run Immediately After a Build Run

1. Ready the Initial Input Unit and the Source Input Unit.
2. Press COMPUTER RESET and then START.

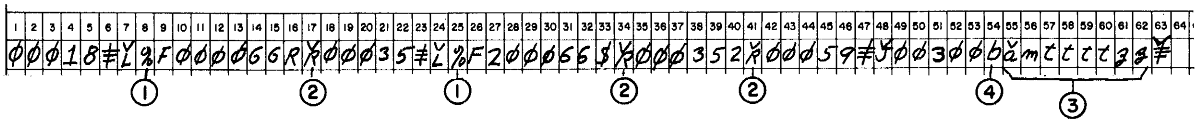
NOTE: If the Bootstrap routine is not on a card in the Initial Input Unit, a Processor Run cannot be initiated after a Build Run.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Ø	Ø	Ø	1	8	‡	=	L	%	F	Ø	Ø	Ø	Ø	6	6	R	=	R	Ø	Ø	Ø	3	5	‡	=	L	%	F	2	Ø	Ø	Ø	6	6	$	=	R	Ø	Ø	Ø	3	5	2	=	R	Ø	Ø	Ø	5	9	‡	=	J	Ø	Ø	3	Ø	Ø	b	=	a	m	t	t	t	t	z	z	=	‡	
								①										②									①										②								②														④		③										

Notes: 1. If SYSAREA 1 is on Channel 2, replace "%" with " ⌑ "
2. If SYSAREA 1 is on Channel 2, replace "R" with "X"
3. a = Ø
m = module number
tttt = first track of SYSAREA1
zz = HA2 of SYSAREA1
4. b = blank

Figure 82. Bootstrap Load Routine Card Image



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Ø	Ø	Ø	1	8	‡	Ľ	%	F	Ø	Ø	Ø	Ø	6	6	R	Ř	Ø	Ø	Ø	3	5	‡	Ľ	%	F	2	Ø	Ø	Ø	6	6	$	Ř	Ø	Ø	Ø	3	5	2	Ř	Ø	Ø	Ø	5	9	‡	J̌	Ø	Ø	3	Ø	Ø	b	ǎ	m	t	t	t	t	z	z	‡̌	
							①									②								①									②							②													④	③									

Notes: 1. If SYSAREA 1 is on Channel 2, replace "%" with " ⌑ "
2. If SYSAREA 1 is on Channel 2, replace "R" with "X"
3. a = Ø
m = module number
tttt = first track of SYSAREA1
zz = HA2 of SYSAREA1
4. b = blank

Figure 83. Bootstrap Routine, Console Entry

HALTS, MESSAGES, AND CORRECTIVE ACTIONS

This section of the manual lists all the halts and messages (and associated corrective actions to be initiated by the machine operator) that can occur during the various types of POS runs, and runs of object programs produced by the POS. The list is divided into two sections: a list of halts, messages, and corrective actions for POS runs, and a similar list for object-program runs.

Identifying Message Numbers

Identifying numbers are composed of five digits, the significance of which is explained below:

Ten-thousands position
- "0" -- dead end waiting loop or halt
- "1" -- message without a waiting loop or halt
- "2" -- waiting loop or halt with one corrective option
- "3" -- waiting loop or halt with two corrective options
- "5" -- waiting loop or halt with four corrective options
- "9" -- waiting loop or halt with any number of corrective options

Thousands position
- "0"

Hundreds position
- "1-2" -- message from IOCS
- "5" -- message from POS
- "7" -- message from Autocoder Processor
- "8" -- message from RPG Processor

Tens and Units position

These positions are used for an identifying sequence number.

Halts, Messages and Corrective Actions for POS Runs

The tabulation of halts, messages and corrective actions for POS runs are arranged in numerical sequence based upon all five digits of the identifying message number.

Operator Options for IOCS Error Conditions During POS Runs

Many of the IOCS error messages provide for more than one corrective procedure. The desired procedure is chosen by pressing INQUIRY REQUEST, entering the appropriate code word via the console, and pressing INQUIRY RELEASE. The corrective actions represented by the code words are as follows:

RETRY -- The IOCS will reattempt to execute the operation that caused the error.

SKIP -- The IOCS will ignore the operation that caused the error and will process the next record or block of records.

PROC -- The IOCS will ignore the error and resume processing as if the operation had been executed successfully.

NOTE: Several of the error messages include the I/O instruction that resulted in the error condition. Error messages for read-tape operations also include the length of the record in which the error occurred. These parts of the message are designated in the following list by "(I/O Op)" and "(R/L)."

Operator Options for IOCS Console Inquiries During POS Runs

The two following options are offered. The desired choice is made by entering the appropriate code word (or address) and pressing the INQUIRY RELEASE key.

START -- The IOCS will restore the status of the indicators and return control to the instruction that was interrupted by the console inquiry.

XXXXX -- "XXXXX" can be any valid storage address (indexing is permitted). The IOCS will branch to this location.

Messages and Explanation	Corrective Action
00510 SYSTAP WLR -- RUN ENDED A wrong length record has been detected on the System Tape. (The tape may also be damaged.)	The job must be restarted.
00511 PATCH TAPE WLR -- RUN ENDED A wrong length record has been detected on the tape from which update data are being read. (This tape may also be damaged.)	The job must be restarted.
00513 OBJ PROGRAM ON DISK FILST, FILND HALT OR ENTER, HLTENT? The object program is in disk storage between FILST and FILND. FILST is the disk address of the first track and FILND is the disk address of the last track.	1. Press INQUIRY REQUEST 2. Enter at the console: ENT (to proceed to the next job) or HLT (to halt) 3. Press INQUIRY RELEASE
00514 ENTER I/O UNIT ASSIGNMENTS The Update Program is requesting the I/O units description for the new System Tape and for the unit from which update data are to be read.	1. Press INQUIRY REQUEST 2. Enter the X-control field for the new System Tape followed by a comma. 3. Enter the X-control field for the unit containing the update data. 4. Press INQUIRY RELEASE.
00519 DISK LOAD END OF JOB The System Load Program has run successfully.	If the user desires to start another run, see "Compound Runs" at the beginning of this section.
10100 NR (I/O Op) Input/Output device is not ready.	Place device in Ready status. (Program will automatically resume when device is Ready.) NOTE: If this message is given because no tape unit is set to the given unit number, the tape unit must not be in a Ready status while the dial is being set to the correct number.
10118 SKC A seek error has been detected.	IOCS will retry the operation. If this error persists, the 7631 File Control should be inspected for a detailed indication of the error.
10505 RUN CARD PRESENT The EOJ card is missing from the source program.	No action is required. The program will continue. NOTE: If any CONFG cards were associated with the job referred to by this message, the parameters on the CONFG cards will not be applied to this job.

Messages and Explanation	Corrective Action
10506 EOF CONDITION The EOJ card is missing from the last source program.	No action is required. The program will continue.
10509 UPDATE COMPLETE End-of-job message.	If the user desires to load the updated POS into disk storage: 1. Press COMPUTER RESET. 2. Press START.
10512 IMPROPER INQUIRY RESPONSE The previous inquiry was answered incorrectly.	1. Press INQUIRY REQUEST. 2. Enter the proper response. 3. Press INQUIRY RELEASE.
10515 SEQ ER xxxx nnnnn Card xxxx in Library section nnnnn is out of sequence.	The program will continue. The card will be internally sequenced by the System Load Program.
10516 UNIDENTIFIED INPUT Unidentified data is encountered by the System Load Program.	Check the Start instruction for proper operands, and the control cards for proper entries, and restart.
10518 EX CD IGNORED An execute card was encountered during the System Load Program.	The program will continue.
10524 UNIDENTIFIED SYSTEM FOR CREATE An unidentified system program has been requested during a Build Run.	No action is required. The program will proceed, ignoring the request.
10529 NO CTL CD -- CORRECT AND PRESS START Current card is not the required CTL card.	Press START to read the next card. (Each time START is pressed, the message will be repeated until a CTL card is encountered.)
10531 SYSAREA 1 STARTS ON TRACK xxxx. ENDS ON TRACK xxxx Message gives new disk area limits of updated system at end of a Disk Update Run.	No action required the program will continue.
10532 INVALID MACHINE SIZE SPECIFIED IN CONFG CARD CARD-NOT PROCESSED Self-explanatory produced during an UPDAT CONFG run. The previous OBJSIZE or MACHSIZE is retained.	No action required.
10533 UPDATE COMPLETE DISK Update EOJ Message	No action required.
10534 EX CD IGNORED An execute card other than a set wordmark card has been encountered and bypassed.	No action required.

Messages and Explanation	Corrective Action
10535 TRYING TO LOAD PAST END LOCATION-RUN DISCONTINUED.	
10590 FORTRAN PROCESSOR NOT AVAIL The FORTRAN Processor requested is not on the System Tape.	No action is required. The request is ignored and the program continues.
10591 COBOL PROCESSOR NOT AVAIL The COBOL Processor requested is not on the System Tape.	No action is required. The request is ignored and the program continues.
10592 COBOL MACROS NOT AVAIL The COBOL section requested of the library is not on the System Tape.	No action is required. The request is ignored and the program continues.
10701 (JOB Card Image) This informs the operator that the source program specified in the Job Card is being run.	No action is required. The program continues.
10704 EOJ -- LITERAL TABLE EXCEEDED The program being assembled contains more incomplete EQU and literals than allowed (see Figure 30).	Run is terminated and the next job (if any) is started. User should rearrange his EQU statements so they are defined as they are processed and/or reduce the number of literals used.
10705 EOJ -- EQU TABLE EXCEEDED The program being assembled contains more than 120 incomplete EQU statements at one time (see Figure 30).	The run is terminated and the next job (if any) is started. User should rearrange EQU statements so that they are defined as they are processed.
10706 EOJ -- MACHINE OR PROGRAM ERROR xxxxx WLR (xxxxx is the name of a macro-instruction.) The block-size count contained in the library does not agree with the computed size.	The run is terminated and the next job (if any) is started. Processing will continue. The user should carefully check all usage of macro-instruction xxxxx.
10801 END RPG This indicates the end of compilation and the beginning of assembly. (This message will not occur if the user has taken the option to terminate the run without assembling.)	No action is required. Processing will continue.
10802 EOJ -- NO RG CONTROL CARD The RG Control Card has been omitted from the RPG specification deck.	Processing is terminated and the next job (if any) is started. Insert the required control card and restart this program.

Messages and Explanation — Corrective Action

10803 EOJ -- NO 1405 CONTROL CARD
The 1405 Control Card has been omitted from the RPG specification deck.

Processing is terminated and next job (if any) is started. Insert the required control card and restart this program.

10804 EOJ -- NO 1301 CONTROL CARD
The 1301 Control Card has been omitted from the RPG specification deck.

Processing is terminated and the next job (if any) is started. Insert the required control card and restart this program.

10805 EOJ -- ERRONEOUS RG CARD
The RG Control Card has erroneous entries. Possible error sources are:

1. Data file not specified in column 7.
2. Wrong mode specified in column 65.
3. Column 8 may specify wrong channel.

Processing is terminated and the next job (if any) is started. Insert the required control card and restart the RPG Processor Run.

10806 EOJ -- ERRONEOUS 1301 CARD
The 1301 Control Card has erroneous entries. Possible error sources are:

1. No record length specified in columns 6-9.
2. No disk channel specified in columns 42-43.
3. No address length or control data field specified in columns 30-31.
4. For Modes 1, 2, 5, 6:
   a. No "F" in column 5.
   b. Disk address or control-data-field lengths not specified in columns 19-20.
   c. Distance between consecutive disk addresses or control data fields not specified in columns 21-23.
   d. Units position of first address or control-data field not specified in columns 24-26.
5. For Modes 3 and 4:
   a. No block length specified in columns 10-13.
   b. No blocking factor specified in columns 14-17.
6. For Modes 5 and 6:
   a. No record length specified in columns 40-41.
   b. No next-record location specified in columns 36-39.

Processing is terminated and the next job (if any) is started. Insert the required control card and restart this program.

10807 EOJ -- NO C OR * IN COL 1 OF INPUT SPEC
Either a "C" or an "*" card is missing from the input specifications in the RPG specification deck.

Processing is terminated and the next job (if any) is started. Insert the required control card and restart this program.

10808 EOJ -- NO * CARD BEFORE FORMAT SPECS
An asterisk card has been omitted from the RPG specification deck.

Processing is terminated and the next job (if any) is started. Insert the required control card and restart this program.

Messages and Explanation	Corrective Action
10809 NO OUTPUT SPECIFIED ON LINE FORMAT SPEC The output has not been specified in the format specifications in the RPG packet.	Processing is terminated and the next job (if any) is started. Insert the required information on the format specification cards, and restart this program.
20114 DCK (I/O Op) Data Check error condition on a write-tape operation. (The IOCS first back-spaced the tape and attempted to rewrite the record, but the error persisted. The IOCS then performed a backspace-skip-rewrite sequence twenty times. The record could still not be successfully written.)	The only possible option for a tape output error is to attempt the write operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20115 LLC (I/O Op) The last line printed or the last card punched contained an error.	The IOCS cannot reprint or repunch records containing an error. To resume program execution, press INQUIRY REQUEST and the INQUIRY RELEASE. (No code word is necessary.)
20116 DCK (I/O Op) Data Check error condition on a unit-record operation. (The IOCS has tried the operation twice, but the error persists.)	The only possible option is to attempt the operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20117 ZRL (I/O Op) Zero record length. (The first character of an output area for a write-tape operation is a group mark with word mark.)	PROC
20143 STK (I/O Op) No Transfer error on a read-card operation. (This is a programming error in the object program. See the 1410 machine manual.)	To cause the IOCS to ignore the error and continue processing, press INQUIRY REQUEST and then INQUIRY RELEASE.
20144 WLR (I/O Op) Wrong-Length-Record error condition on a unit-record operation.	The only possible option is to try the operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20183 CI (*) The last information entered through the console was invalid or cancelled by the operator. (*) In some cases, the information written in this area of the previous IOCS message (for example, "(I/O Op)") is still in storage and is rewritten with this message. This can be helpful in locating the previous IOCS message in order to determine what the correct console entry should have been.	Press INQUIRY REQUEST, enter the correct information, and then press INQUIRY RELEASE.

Messages and Explanation.	Corrective Action
20501 (No Indicative Message) The Bootstrap Program has failed to load properly.	1. Restart the job in its entirety.
20502 RUN CARD ERROR (Label) An invalid RUN card (identified by "Label") is included in the source program.	1. Press INQUIRY REQUEST 2. Enter the proper label at the console. 3. Press INQUIRY RELEASE.
20503 CONFG OPR ERR (Label) A CONFG card (identified by "Label") with an invalid operand has been detected.	1. Press INQUIRY REQUEST. 2. Enter the proper operand at the console. 3. Press INQUIRY RELEASE.
20507 OBJECT PROGRAM TAPE REQUIRES NEW REEL. PRESS START TO CONTINUE. The rewind-after-each-object-program option has been requested.	1. Mount a new tape reel. 2. Press START.
20517 ILLEGAL INPUT CARD The format of the card being processed is not acceptable to the System Load Program.	1. Display symbolic location "INAREA." 2. Correct any format errors. 3. Branch to symbolic location "PROXX."
20520 xxxxxxxxxx yyyyy CONFG card with label xxxxxxxxxx has an improper yyyyy ("LABEL" or "OPRND") field.	1. Press INQUIRY REQUEST. 2. At the console enter the proper label or operand. 3. Press INQUIRY RELEASE.
20522 CONFG LAB ERR (Label) A CONFG card (identified by Label) with an improper label has been encountered.	1. Press INQUIRY REQUEST. 2. Enter the proper label at the console. 3. Press INQUIRY RELEASE.
20523 ENT CONFG OPR (Label) This message will follow message "20522."	1. Press INQUIRY REQUEST. 2. Enter the proper operand at the console. 3. Press INQUIRY RELEASE.
20525 RUN OPR ERR (Operand) A RUN card with an invalid operand has been encountered.	1. Press INQUIRY REQUEST. 2. Enter the correct operand at the console. 3. Press INQUIRY RELEASE.
20526 NO RUN CARD A RUN card has been omitted from the source program.	1. Press INQUIRY REQUEST. 2. Enter the RUN card label at the console. 3. Press INQUIRY RELEASE.
20527 EXAMINE DECK SETUP-RUN ENDED The update data are incorrectly arranged, or a card is mispunched.	1. Correct the update data. 2. Restart the Update Run.

Messages and Explanation	Corrective Action
20530 NO CTLCD-CORRECT AND PRESS START The first card after the SYSTEM RUN card for a Disk Update run is not an UPDAT, INSER, or DELET card.	Restart the run or reload the deck until a control card first and press start.
30181 HLT xxxxx The IOCS is ready to accept information through the console. All channels are free ("xxxxx" is the Resumption Address).	1. START 2. XXXXX
30500 LD-UD Either the System Load Program or the Update Program is to be run.	1. Press INQUIRY REQUEST. 2. Enter at the console: "LD" for System Load Program, "UD" for Update Program. 3. Press INQUIRY RELEASE.
30508 LISTING ON DISK FILST, FILND. PROCEED OR ENTER. PRO-ENT The object program listing is in disk storage between FILST and FILND. FILST is the disk address of the first track and FILND is the disk address of the last track.	1. Press INQUIRY REQUEST. 2. Enter at the console: PRO (to proceed with the output), ENT (to start the next job.) 3. Press INQUIRY RELEASE.
30521 aaaaa EOF n x A processor has attempted to use a track which is not within a defined system area. aaaaa -- address of file n -- System Area of file x -- "I", if input file, or "O" if output file. For input files this message can be the result of an invalid format of the System Areas. For output files this message is usually caused when not enough cylinders are defined for the System Areas. If the condition occurs in System Area 1, increase this area. If the condition occurs in System Area 2, 3, or 4, increase each of these three areas. (See Cylinder Requirements in section on Detailed Machine Requirements and Considerations.)	If the user has more jobs to run in this batch: 1. Press INQUIRY REQUEST. 2. Enter CONT at the console. 3. Press INQUIRY RELEASE.
30599 EOB End of Batch	To start another batch of jobs: 1. Place the data into the main input unit. 2. Place CONFG cards (if any) into the Initial Input Unit. 3. Do <u>not</u> enter the Bootstrap program. (It is not required.) 4. Press INQUIRY REQUEST. 5. Enter "CONTb" at the console. 6. Press INQUIRY RELEASE.

Messages and Explanation	Corrective Action
40110 DCK (I/O Op) (R/L) Data Check error condition on a read-tape operation. (The IOCS attempted to read the record twenty times, but the error persisted.)	1. RETRY 2. SKIP 3. PROC
40117 CHC Format Character Check Check Character Code Check Parity Check Write Disk Check	1. SKIP 2. PROC 3. REDO
40528 NO END CARD There is no END card in the Update Program.	1. If the user is duplicating the System Tape only and desires to continue: a. Press INQUIRY REQUEST. b. Enter COPY at the console. c. Press INQUIRY RELEASE. 2. If the user desires to stop the run and rewind all tapes: a. Press INQUIRY REQUEST. b. Enter HALT at the console. c. Press INQUIRY RELEASE. 3. If the user desires to continue the Update Run: a. Press INQUIRY REQUEST. b. Enter CONT at the console. c. Press INQUIRY RELEASE.
40150 CND Wrong Length Format, or File Control Circuit	1. SKIP 2. PROC 3. REDO
40151 NRF No Record Found, or Disk Storage Circuit Check (See RPG CONSIDERATIONS also.)	1. SKIP 2. PROC 3. REDO
40152 ITN Invalid Track Number	1. SKIP 2. PROC 3. REDO
40153 WLR Wrong-Length Record	1. SKIP 2. PROC 3. REDO
40182 HLT xxxxx The IOCS is ready to accept information through the Console. All Channels are free. ("xxxxx" is the Resumption Address.)	1. START 2. XXXXX

Messages and Explanation	Corrective Action
xxxxx OPERAND # yyy MISSING Required parameter yyy is missing from macro-instruction xxxxx.	Compilation continues. User should check usage of macro xxxxx.
P/L xxxxx (Message) This is the format of messages produced by the diagnostic portion of the library subroutines, written by the user. xxxxx is the name of the macro-instruction that calls in the subroutine.	The action to be taken (if any) is specified by the user's subroutine.

COBOL Processor Run Messages (Console)

The following messages do not cause the processor to stop compilation:

1. BEGIN COBOL RUN
2. (name-of-job) CB-NNNN (identification field)
   Note: NNNN is the version and modification number which will be supplied by the compiler.

Each of the following messages will terminate compilation; the output from the job concerned will be produced and the processor will continue with the next job.

1. IMPROPER ID DIVISION, RESTART COBOL RUN. Action - job must be rerun after insuring that the IDENTIFICATION (ID) DIVISION card immediately follows the COBOL RUN card.
2. INSERT PROGRAM-ID PARAGRAPH. RESTART COBOL RUN. Action - job must be rerun after insuring that the PROGRAM-ID paragraph immediately follows the ID DIVISION Card.
3. RUN DISCONTINUED -- SEE DIAGNOSTIC
   Action - job must be rerun after source program has been corrected.

The following message may be produced by an uncorrectable error, (machine failure, undetected source program error, etc.)

COBOL HALT XXXXX/YYYYY, X1 = ZZZZZ
DUMP or PRESS START FOR OUTPUT

Note: XXXXX is the name of phase during which the error occured.
YYYYY is the address of the last instruction executed in the phase.
ZZZZZ is the contents of X1.

After the message the processor halts. At this point the operator has the option of either:

1. taking a MEMORY DUMP
2. Pressing START to obtain the output from the job concerned and to continue with the next job.

FORTRAN Processor Run Messages (Console)

NO BOP CARD
Run cards out of card reader. Load - BOP - Card. Reload cards in card reader and ready it. Press start to continue.

SUBPROGRAM NAME DIFFERS FROM BOP CARD
After the halt, correct the BOP card, reload and press start to continue. Proceeding without correction (press start) will result in the end card having an improper branch and prior to executing or assembling this must be corrected. To discontinue the run and start another job, the system must be readied manually by rewinding tapes, keying in load program, etc.

EXCEEDED ARRAY TABLE SIZE
This message will be printed on the console if the source program contains too many different array names. The source program must be revised if this diagnostic message occurs. Run terminates. To start another job the system must be readied manually.

END OF BATCH
If input to Autocoder was specified, the Autocoder run will begin. Otherwise, the system will come to a halt.

ILLEGAL OUTPUT OPTION SPECIFIED
The run will terminate.

Halts, Messages and Corrective Actions for Object Program Runs

The tabulation of halts, messages and corrective actions for object program runs are arranged in numerical sequence of the three low-order digits of the identifying message numbers.

The RPG object program messages, which do not have identifying message numbers, are listed in alphabetical order.

Operator Options for IOCS Error Conditions During Object Program Runs

Many of the IOCS error messages provide for more than one corrective procedure. The desired procedure is chosen by pressing INQUIRY REQUEST, entering the appropriate code word via the console, and pressing INQUIRY RELEASE. The corrective actions represented by the code words are as follows:

RETRY --- The IOCS will reattempt to execute the operation that caused the error.

SKIP -- The IOCS will ignore the operation that caused the error and will process the next record or block of records.

PROC -- The IOCS will ignore the error and resume processing as if the operation had been executed successfully.

ACCEPT-- The IOCS will ignore the error. (This option is offered for errors caused by tape labels.)

DUMP -- The IOCS will write the record that contains the error onto the output error file. After writing the record on that file, the IOCS will again enter a waiting loop and write the same error message, enabling the operator to continue processing with another option. (This option is offered only if an output error file was designated by a DIOCS "READERROR" entry containing both the "SCAN" and "TAPE, #2, 1Y" operands.)

*SCAN -- The IOCS will type the location(s) of the asterisk(s) in the record. After typing the location(s), the IOCS will again enter a waiting loop and write the same message, enabling the operator to continue processing with another option. (This option is offered only if the DIOCS "READERROR" entry contains the operand "SCAN.")

NOTE: Several of the error messages include the I/O instruction that resulted in the error condition. Error messages for read-tape operations also include the length of the record in which the error occurred. These parts of the message are designated in the following list by "(I/O Op)" and "(R/L)."

Operator Options for IOCS Console Inquiries During Object Program Runs

The following three options are offered when an inquiry request is made from the console and no support for the DIOCS "INQUIRY" entry was provided in the program. (That is, the programmer has not written a routine that processes console inquiries.) The options are chosen by entering the code word (or address) and pressing the INQUIRY RELEASE key.

START -- The IOCS will restore the status of the indicators and return control to the instruction that was interrupted by the console inquiry.

CHKPT -- The IOCS will cause a checkpoint to be taken and will then return to the inquiry-request waiting loop. The operator then has the choice of terminating the run or resuming it (by using the START option).

XXXXX -- "XXXXX" can be any valid storage address (indexing is permitted). The IOCS will branch to this location.

List of Halts, Messages and Corrective Actions --
Object Program Runs

Messages and Explanation	Corrective Action
IOCS Messages	
00255 ARM No access arm is available	The program cannot proceed.
10100 NR (I/O Op) Input/Output device is not ready.	Place device in Ready status. (Program will automatically resume when device is Ready.) NOTE: If this message is given because a tape unit is set to a wrong number, the tape unit must not be in a Ready status while the dial is being set to the correct number.
10111 DCK (I/O Op) (R/L) Data Check error condition on a read-tape operation. (The IOCS attempted to read the record twenty times, but the error persisted. The IOCS has written the record on the output error file, as specified by the operand of the DIOCS "READERROR" entry.)	None. (The IOCS does not enter a waiting loop for operator action.)
10118 SKC A seek error has been detected.	IOCS will retry the operation. If this error persists, the 7631 File Control should be inspected for a detailed indication of the error.
10125 EOF Sequential end-of-file has occurred. This is an informative message. There is no halt.	The program will continue.
10185 CPT ccc The IOCS has taken a checkpoint. "ccc" is the accumulated count of checkpoints taken since the start of the program.	None. (The IOCS does not enter a waiting loop for operator action.)

10254 xyz
x, y, or z can be "0" or "1." If "0" then the error type is not applicable. Error types are as follows:
x = No Transfer
y = Busy
z = Wrong Length Record
	The program will automatically reattempt execution of the I/O instruction.

10257 auc
Where:
a = identifying arm number
u = identifying unit number
c = identifying channel number
The program has removed the arm identified above from the list of available arms.
	The program will automatically resume operation with the available arms.

Messages and Explanation	Corrective Action
20258 OPN xxxxx GET An invalid disk address has been encountered starting at location xxxxx.	The operator must: 1. Change the information starting at location xxxxx by performing the standard ALTER procedure. 2. Press INQUIRY REQUEST. 3. Press INQUIRY RELEASE. 4. The program will resume processing.
10314 TIE (cu) bc rc ht (*) bc rc ht (**) Trailer Label-In-Error. "bc" is the block count, "rc" is record count, "ht" is the hash total. Record counts and hash totals are given only if specified by the DIOCS "COUNTS" entry. Block counts are always given. (*) This line is taken from the trailer label. (**) This line is accumulated by the IOCS.	None. (The IOCS does not enter a waiting loop for operator action.)
20114 DCK (I/O Op) Data Check error condition on a write-tape operation. (The IOCS first back-spaced the tape and attempted to rewrite the record, but the error persisted. The IOCS then performed a backspace-skip-rewrite sequence twenty times. The record could still not be successfully written.)	The only possible option for a tape output error is to attempt the write operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20115 LLC (I/O Op) The last line printed or the last card punched contained an error.	The IOCS cannot reprint or repunch records containing an error. To resume program execution, press INQUIRY REQUEST and the INQUIRY RELEASE. (No code word is necessary.)
20116 DCK (I/O Op) Data Check error condition on a unit-record operation. (The IOCS has tried the operation twice, but the error persists.)	The only possible option is to attempt the operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20117 ZRL (I/O Op) Zero record length. (The first character of an output area for a write-tape operation is a group mark with word mark.)	PROC
20120 EOR (cu) Input or output end-of-reel condition. (This message is given only for files that have not been assigned an alternate tape unit.)	Mount the next reel, press INQUIRY REQUEST then INQUIRY RELEASE.

Messages and Explanation	Corrective Action
20136 RLN The RDLIN macro-instruction was executed, but the card read by the IOCS was not recognized as a label card (i.e., columns 16-30 did not contain "RDLIN").	Place a label card in the card reader. Press INQUIRY REQUEST and then INQUIRY RELEASE. (The RDLIN will be re-executed.)
20143 STK (I/O Op) No Transfer error on a read-card operation. (This is a programming error in the object program. See the 1410 machine manual.)	To cause the IOCS to ignore the error and continue processing, press INQUIRY REQUEST and then INQUIRY RELEASE.
20144 WLR (I/O Op) Wrong-Length-Record error condition on a unit-record operation.	The only possible option is to try the operation again. This option is assumed by the IOCS if the operator presses INQUIRY REQUEST and then INQUIRY RELEASE. (No code word is necessary.)
20183 CI (*) The last information entered through the console was invalid or cancelled by the operator. (*) In some cases, the information written in this area of the previous IOCS message (for example, "(I/O Op)") is still in storage and is rewritten with this message. This can be helpful in locating the previous IOCS message in order to determine what the correct console entry should have been.	Press INQUIRY REQUEST, enter the correct information, and then press INQUIRY RELEASE.
20186 RST The IOCS has made the program ready for restarting (from a checkpoint). This waiting loop allows the operator to perform additional "setups," if any are required.	To begin execution at the restart point, press INQUIRY REQUEST and then INQUIRY RELEASE.
20259 INVALID PUT AT xxxxx A PUT has been attempted on a record which does not exist. xxxxx is the address of the invalid record address.	The operator must: 1. Press INQUIRY REQUEST. 2. Press INQUIRY RELEASE. 3. The program will continue processing, bypassing the invalid PUT.
20260 SKC SEEK Error	The operator must: 1. Press INQUIRY REQUEST. 2. Press INQUIRY RELEASE. 3. The program will continue, repeating the SEEK operation.

Messages and Explanation	Corrective Action
30132 FIL (cu) (*) (**) The header label indicates that this is not the correct input tape. (*) The file serial number, reel sequence number, file name, and the creation date from the header label. (**) The above information as specified by the DTF entries.	1. RETRY (After mounting the correct input tape.) 2. ACCEPT
30133 NIH (cu) No input header label found by the IOCS.	1. RETRY 2. ACCEPT (This means the IOCS will accept the first tape record as a "false" header label and treat the next tape record as the first data record. Therefore, if the tape does not actually contain a header label, the operator must rewind the tape to load point.)
30181 HLT xxxxx or 40182 HLT xxxxx (*) The IOCS is ready to accept information through the console. All channels are free. ("xxxxx" is the Resumption Address.) (*) This message is given if a DIOCS "CHECKPOINT" entry was written for the program.	1. START 2. CHKPT (This option can be used only when the 40182 message is given.) 3. XXXXX

30256 xyz
x, y, or z can be "0" or "1." If "0," then the error type is not applicable. Error types are as follows:
x = No Transfer
y = Condition
z = Data Check
	The operator can have the IOCS skip or process the record involved:
	1. Press INQUIRY REQUEST
	2. Enter SKIP or PROC at the console
	3. Press INQUIRY RELEASE
	4. The program will continue
	Notes:
	1. The user may alter the disk address prior to entering SKIP or PROC by performing the standard ALTER procedure with the following limitations:
	(a) Addresses cannot be changed if sequential processing is used.
	(b) Address changes must be within the same channel.
	2. PROC cannot be used with a No Transfer error ("1" in the "x" position).

30257 PUT This indicates a logic error in the issuing of the PUT macro-instruction in the user-written program.	None.
40110 DCK (I/O Op) (R/L) Data Check error condition on a read-tape operation. (The IOCS attempted to read the record twenty times, but the error persisted.)	1. RETRY 2. SKIP 3. PROC

Messages and Explanation	Corrective Action
40117 CHC Format Character Check Check Character Code Check Parity Check Write Disk Check	1. SKIP 2. PROC 3. REDO
40130 NOH (cu) No output header label found by the IOCS.	1. RETRY 2. ACCEPT (This means the IOCS will accept the tape as unlabeled and therefore cannot check the date or retention cycle.) 3. The operator may key the date through the console (yyddd format, where yy is the year and ddd is the day). The IOCS will move the date to storage positions 115-119 and will automatically RETRY the header label check. (Although this option is not a corrective procedure, it is offered to enable the operator to store the current date through the console, in the event the date was not loaded by card. This option also insures that the new output header labels will have the correct date.)
40131 DAT (cu) Date or retention cycle on the header label of an output tape indicates the present tape records should not be destroyed.	1. RETRY (After mounting a "scratch" tape.) 2. ACCEPT 3. The same as option 3 for the preceding message (40130 NOH). (This option is offered to cover the possibility that the absence of a date in locations 115-119 caused this message to be given. It also prevents the creation of header labels with blank date fields. Tapes with such a label will always be treated by the IOCS as "scratch" tapes.)
40150 CND Wrong Length Format, or File Control Circuit Check.	1. SKIP 2. PROC 3. REDO
40151 NRF No Record Found, or Disk Storage Circuit Check.	1. SKIP 2. PROC 3. REDO
40152 ITN Invalid Track Number.	1. SKIP 2. PROC 3. REDO

Messages and Explanation	Corrective Action
40153 WLR Wrong-Length Record.	1. SKIP 2. PROC 3. REDO
50112 DCK (I/O Op) (R/L) Data Check error condition on a read-tape operation. (The IOCS attempted to read the record twenty times, but the error persisted.)	1. RETRY 2. SKIP 3. PROC 4. *SCAN
60113 DCK (I/O Op) (R/L) Data Check error condition on a read-tape operation. (The IOCS attempted to read the record twenty times, but the error persisted.)	1. RETRY 2. SKIP 3. PROC 4. *SCAN 5. DUMP

RPG Object Program Messages

In addition to any IOCS messages, the following messages can occur during the run of an object program produced by the RPG Processor:

Messages and Explanation	Corrective Action
CHAIN ADD ERR The chain address is in error.	Press START to bypass this entire record.
CTL DATA CHECK Control data check error	Press START to bypass the record.
INPUT REC OUT OF SEQ Data is not in the proper order.	Sequence the data and restart the object program run.
RECORD TYPE NOT FOUND Data of the type specified for the report is missing.	Supply the correct data and restart the entire object program run.

COBOL Object Program Messages

In addition to any IOCS messages, the following messages can occur during the run of an object program produced by the COBOL processor:

When using ACCEPT FROM CONSOLE WITH PRIORITY, the following messages are given:

1. PRESS INQUIRY RELEASE.
2. PRESS INQUIRY REQUEST.

Action: Pressing Inquiry Release will restart program; when Press Inquiry Request appears, do so, then type in data required and press Inquiry Release to restart program.

FORTRAN Object Program Messages

ER 1

With a negative argument in the square root routine, the sign is changed to plus prior to taking the square root of that floating point number.

ER 2

Improper d modifier in floating arithmetic command. Exit from that routine.

ER 00803

This is an error for invalid exponent in E conversion. Record ignored and no data is transferred.

ER 00804

Nonexistent tape drive specified; e.g., specifying unit 35. Specification ignored. Halt. Program cannot be executed, must recompile.

ER 8

Invalid log. argument. Exit from log. routine.

ER 9

Exponential overflow. Exit from exponential routine.

ER 20

Argument greater than 999 radians in Sine and Cosine routines.

ER 10

Argument out of range in XINT, fixed size smaller than floating size. Exit from the routine.

ER 11

Second parameter in MOD is zero.

Exit from the routine.

* ADDR

Field overflow on F or 1 output

ER 12

Second parameter in XMOD is zero.

Exit from the routine.

Discopen Routine -- Error Messages

ER 00805

Occurs when there aren't the proper disk read, write or find codes in location tape.

ER 00807

This error occurs when there is a Define File N2 error, when there are not enough tracks specified for the job or the maximum value of (i) exceeds this limit.

ER 00808

No track address in the control cards will give this message. Actually the base cylinder is missing for the specified disk statement.

Disk Section of I/O Common

ER 00801

There is an error in either the Define File N1 Section or Zeros in Location 615 to 619 - Address supplied by the processor.

I/O Common Routine

ER 00802

Will occur when there is an Invalid Format supplied by the Programmer.

ER 00806

This is an error in the P Factor section of E Conversion. The error states that the P Factor (P) plus Decimal (D) is greater than Width (W).

XFIX Routine

ER 00810

XFIX message 500> 504 exponent higher than BOP size.

INDEX

ACCEPT Console Entry 57
Advantages POS 5
Additional Core Storage 8
Additional Cylinders 8
Appendix A -- The 1401 Print/Punch Program 27
Appendix B -- CONTROL TABLES 29
Appendix C -- FORTRAN RELOCATABLE PROGRAM LOADER 31
Assembly Option (RPG) 18
Autocoder 16
Autocoder Considerations 16
Autocoder Deck Setup 17
Autocoder Relocation 31
Autocoder Table Limits 17

Batch Compiling 19
Bootstrap Routine 46
Build Run 10
Build Run Deck 14, 15
Build Run -- Operator's Guide 45

Changing the Standard Operating Configuration -- Update Run 23
Card Deck -- Print Run 25
Card Deck -- Update Run 24
Card Labels 21
Card Oriented Systems 10
Channel Number 1301 36
COBOL 5, 10
COBOL Considerations 18
COBOL Object Program Messages 64
COBOL Processor Run 56
COBOL Run Options 18
Communication Table 29, 30
Compound Runs -- Operator's Guide 46
CONFG Cards 11
CONFG-Card Types 12
Console Inquiries -- IOCS Errors, POS Runs 47
Console Inquiries -- Object Program Runs 57
Control Cards -- Build Run 10
Control Cards -- FORTRAN 34
Control Cards -- Print/Punch Program 27
Control Cards -- Print Run 24
Control Cards -- Processor Runs 16
Control Card and Sense Switch Options 27
Control Cards -- Update Program 25
Control Cards -- Update Run 21
Control Cards -- Updating the Library 23
Control Tables 29
CREAT Cards 10
CREAT Card Examples 11
Creation of POS -- The Build Run 10
CTL Cards (Autocoder) 17
Cylinder Requirements 5

Deck Setup -- Autocoder 17
Deck Setup -- Print Run 25
Deck Setup -- RPG 18
Deck Setup -- Update Run 26
Definition Cards 20
Detailed Machine Requirements and Considerations 7
Disk Track Requirements 7
Disk Update Program 25, 9
Disk Update Setup 26
Distribution of System Areas 7
DUMP Console Entry 56
Duplicating the System Tape 24

END Card (Autocoder) 16
EOJ Card 16
EOL Card 20
EOP Card 20, 22
EOS Card 20
EOT Card 20
Error Correction -- Print/Punch Program 28
Expanding System Functions 8

Features of POS 5
Final Input Tape 34
Format Track Generation 8
Format Track Requirements 7
FORTRAN 5
FORTRAN Compiler 32
FORTRAN Considerations 19
FORTRAN Deck Assemblies 19
FORTRAN Processor Run Messages (Console) 56
1401 Print/Punch Program 27

General Considerations -- POS 9

Halts, Messages, and Corrective Actions 47
Halts, Messages, and Corrective Actions COBOL 56
Halts, Messages, and Corrective Actions - FORTRAN 56
Halts, Messages, and Corrective Actions -- POS Runs 47
Halts, Messages, and Corrective Actions -- POS Runs - RPG 18
Halts, Messages, and Corrective Actions -- Object Program Runs 57
HDRCHECK CONFG Card 13
Header Cards 21
Header Label Checking 13
Home Address and Record Address Generation 8

Identifying Message Numbers 47
Initial Input Unit 12
INITINPT CONFG Card 12
INPTELEM CONFG Card 12
Input Program 9
INSER Control Cards 24
Intermediate Tape Unit 35
Introduction 5
IOCS Errors -- Object Program Runs 57
IOCS Errors -- POS Runs 47
IOCS Messages -- Object Program Runs 57

Library 41
Library Directory 29
Library Directory Entries 29
Library Format 23
Library Subroutines 41
Library Tape Unit 34
Library Update Control Cards 23

List of Halts, Messages, and Corrective Actions -- Object Program Runs 57
List of Halts, Messages, and Corrective Actions -- POS Runs 48
Listing the Library from the System Tape 23
Load Cards (Autocoder) 17
Load Mode 7
Load Program 10
Logic Descriptions 38
Lower Load Limit 35

Machine Options 6
Machine Requirements 5
Machine Requirements -- Detailed 7, 27
Machine Requirements -- Print/Punch Program 27
MACH SIZE CONFG Card 13
Maintenance and Auxiliary Functions 20
Memory Map 36
Message Numbers 47
Module Available 1301 36
Move Mode 8

Namex 21, 22

Object Machine Size 13
Object Program Messages 57
Object Program Runs -- Halts and Messages 57
Object Tape Assignment 35
OBJSIZE CONFG Card 13
Off-Line Punching and Printing on the 1401 27
Operating Configuration 10
Operating Instructions 43
Operator Guide - Print/Punch Program 27
Operator's Guide -- POS 45
Operations Options -- IOCS Errors -- POS Runs 47
Operator's Options -- IOCS Errors -- Object Program Runs 57
Optional System Programs 11
Options Machine 6
Output Program 9

Phase Directory 29
Phase Entries -- Phase Directory 29
Phase Names - List 21
Phase Update Control Cards 21
POS Runs -- Halts and Messages 47, 48
POS System Area 14
Prerequisites 5
PRINTELEM Configuration Card 12
Print Program 9, 28
Print/Punch Program 28
Print Run 25
Print Run -- Operator's Guide 46
Print Suppression 13
Print Unit 12
PROC Console Entry 29, 38
Processors 9
Processor Machine Size 13
Processor, Print and Disk Update Runs 46
Processor Runs 16
Processor Run -- Operator's Guide 47
Program Entries -- Program Directory 29
Program and Phase Directory 29
Program Update Control Card 20

Programming Considerations 38
PUNCHELEM Configuration Card 13
Punch Suppression 13
Punch Relocated Deck 37
Punch (Unit) 13
Purpose of this Publication 5
Purpose of POS 5

Relocation 14, 38
Relocation Indicators 33
RELOCATE CONFG Card 14
Restrictions (Autocoder) 17
RETRY Console Entry 47
RPG 9
RPG Assembly Option 18
RPG Consideration 17
RPG Run Deck Setup 18
RPG Messages -- Object Program Runs 47
RUN Card 16

SCAN Console Entry 47
Sense Switch Settings 37
SKIP Console Entry 47
Source Input Unit 12
Standard Operating Configuration 10, 23
Standard Operating Configuration -- System Tape Supplies 14
START Card 21
START Card Labels 21
START Console Entry 21, 47
Start Instruction 21
Supervisory Programs 9
SUPPRESS CONFG Card 13
Suppression of Punch and/or Print Output 13
SYSAREA CONFG Card 14
Sysname 23
System Control Program 9, 10
System Functions 8
System Load Program 9
System Library 9
System Programs 9, 21
System Supervisory Programs 9
System Tape 10

Tape Update Program 9
Tape Update Run 45

UPDATE Cards 24, 46
Update Program 9, 24
Updating a Phase 21
Updating a Program 20, 24
Update Run 20, 24, 46
Update Run - Operator's Guide 25
Updating the Library 23
Upper Load Limit 35
Use of Additional Core Storage to Expand System Functions 8
Use of POS -- Prerequisites 5
Using the 1410 in 1401 Mode 28

Working Storage 7
Write Relocated Tape 36
Writing Source Progr ams - Prerequisites 5

IBM Technical Newsletter

File Number	1410-27
Re: Form No.	C28-0287-1
This Newsletter No.	N27-1215
Date	March 18, 1965
Previous Newsletter Nos.	N28-1073 N28-1147 N28-1186

IBM 1410 PROCESSOR OPERATING SYSTEM (DISK)

This Technical Newsletter amends the publication IBM 1410 Processor Operating System Using IBM 1301 Disk Storage, Form C28-0287-1, to add an illustration of the control cards required for inclusion of user routines at POS creation, to provide an additional error message explanation, and to incorporate minor corrections.

Substitute the attached replacement pages (41-42 and 61-62) for the corresponding pages currently in the manual. In addition, the following change should be made.

Page	Change
59	The second message should read: 10134 TIE (cu).

Please file this cover letter at the back of the publication. The letter will provide a reference to changes, a method of determining that all changes have been received, and a check for determining whether or not the user's publication currently contains the proper pages.

IBM Corporation, Programming Systems Publications, Dept. 637, Neighborhood Road, Kingston, N.Y. 12401

PRINTED IN U.S.A.

IBM Technical Newsletter

File Number	1410-27
Re: Form No.	C28-0287-1
This Newsletter No.	N28-1186
Date	November 9, 1964
Previous Newsletter Nos.	N28-1073 N28-1147

IBM 1410 PROCESSOR OPERATING SYSTEM USING IBM 1301 DISK STORAGE

This technical newsletter amends the publication IBM 1410 Processor Operating System Using IBM 1301 Disk Storage, Form C28-0287-1, to expand and clarify information presently in the manual.

The attached replacement pages (contents, 5 through 8, 11 through 14, 39 through 42, and 45 and 46) should be substituted for the corresponding pages currently in the manual. Changes on the replacement pages are indicated by a vertical line to the left of the updated text or by a dot (•) to the left of figure captions.

Please file this cover letter at the back of the publication. The letter will provide a reference to changes, a method of determining that all changes have been received, and a check for determining whether or not the user's publication currently contains the proper pages.

IBM Corporation, Programming Systems Publications, Dept. 637, Neighborhood Road, Kingston, N.Y. 12401

PRINTED IN U.S.A.

Printed in U.S.A. C28-0287-1

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