SPERRY UNIVAC
COMPUTER SYSTEMS

PUBLICATIONS REVISION

Operating System/3 (OS/3)

Emulation/Conversion (9200/9300)

User Guide/Programmer Reference

UP-8063 Rev. 3

This SPERRY UNIVAC® Operating System/3 (OS/3) Library Memo announces the release and availability of "SPERRY UNIVAC Operating System/3 (OS/3) Emulation/Conversion (9200/9300) User Guide/Programmer Reference", UP-8063 Rev. 3. This is a Restricted Distribution Item (RD). Order where necessary.

This document has been revised to expand and update information concerning the concepts and basic steps required for emulator generation and file conversion. The system description form and descriptor cards described in Section 6 have been extensively updated to include new features and enhancements as well as to provide comprehensive diagnostic coverage for the field entries specified on these cards. In addition, a new disc file copy program (COPY$10) and a disc file conversion program (DCON92) have been added to Appendixes A and C, respectively.

Destruction Notice: This revision supersedes and replaces "SPERRY UNIVAC Operating System/3 (OS/3) Emulation/Conversion (9200/9300) User Guide/Programmer Reference" UP-8063 Rev. 2, released on Library Memo dated November 1975. Also destroyed are Updating Package A, UP-8063 Rev. 2–A released on Library Memo dated April 1976 and Updating Package B, UP-8063 Rev. 2–B released on Library Memo dated July, 1976. Please destroy all copies of UP-8063 Rev. 2, UP-8063 Rev. 2–A, UP-8063 Rev. 2–B and their Library Memos.

Additional copies may be ordered by your local Sperry Univac Representative.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists 217, 630 (less 631E, 634, 635A) 692, 75, and 76	Mailing Lists 631E, 634 and 635A (Cover and 222 pages)	Library Memo for UP-8063 Rev. 3
		RELEASE DATE: September 1976

SPERRY UNIVAC
Operating System/3(OS/3)
Emulation/Conversion (9200/9300)

User Guide/Programmer Reference

H

SPERRY UNIVAC
COMPUTER SYSTEMS

UP-8063 Rev. 3

SPERRY UNIVAC
Operating System/3 (OS/3)

Emulation/Conversion (9200/9300)

User Guide/Programmer Reference

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

Sperry Univac is a division of Sperry Rand Corporation.

AccuScan, FASTRAND, PAGEWRITER, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are trademarks of the Sperry Rand Corporation.

PRINTED IN U.S.A.

PAGE STATUS SUMMARY

ISSUE: UP-8063 Rev. 3

Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level	Part/Section	Page Number	Update Level
Cover								
Title Page								
PSS	1							
Preface	1							
Contents	1 thru 6							
PART 1								
Title Page								
1	1 thru 7							
2	1 thru 4							
3	1 thru 6							
PART 2								
Title Page								
4	1 thru 6							
PART 3								
Title Page								
5	1 thru 5							
6	1 thru 65							
7	1, 2							
PART 4								
Title Page								
Appendix A	1 thru 20							
Appendix B	1 thru 16							
Appendix C	1 thru 66							
Appendix D	1 thru 3							
Index	1 thru 7							
User Comment Sheet								

All the technical changes are denoted by an arrow (→) in the margin. A downward pointing arrow (↓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (↑) is found. A horizontal arrow (→) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is one of a series designed to instruct and guide the programmer in the use of the SPERRY UNIVAC Operating System/3 (OS/3). This manual specifically describes OS/3 9200/9300 Emulation/Conversion and its effective use. Its intended audience is the programmer, system analyst, or manager of a SPERRY UNIVAC 9200 or 9300 system. The expression 9200/9300 is used to denote either system.

Another manual that covers OS/3 Emulation/Conversion is the introductory manual. The introductory manual briefly describes OS/3 Emulation/Conversion and its facilities.

This user guide is divided into the following parts:

- PART 1. EMULATION

  Introduces emulation and conversion in terms of what they are and the way they are used. Describes various considerations which should be examined prior to emulation. Describes preparatory steps and methods of access as well as approaches to emulation.

- PART 2. CONVERSION

  Introduces conversion in terms of the problems involved.

- PART 3. EMULATOR GENERATION

  Describes how to generate a SPERRY UNIVAC 9200/9300 emulator.

- PART 4. APPENDIXES

Contents

PAGE STATUS SUMMARY

PREFACE

CONTENTS

PART 1. EMULATION

1. INTRODUCTION TO EMULATION AND CONVERSION

1.1. GENERAL 1—1

1.2. HOW EMULATION WORKS 1—3

1.3. WHAT WILL AND WILL NOT BE EMULATED 1—6

2. PLANNING FOR EMULATION

2.1. GENERAL CONSIDERATIONS PRIOR TO EMULATION 2—1
2.1.1. Data Files 2—2
2.1.2. Transferring Programs 2—3
2.1.3. Creating Emulators 2—4

3. SYSTEM PREPARATION

3.1. PREPARATORY STEPS 3—1

3.2. METHODS OF ACCESS 3—1

3.3. SYSTEM GENERATION 3—2
3.3.1. File OS/3 JCL Only 3—3
3.3.2. File OS/3 JCL and System Elements 3—3

PART 2. CONVERSION

4. FILE AND PROGRAM CONVERSION

4.1. GENERAL 4—1

4.2. FILE CONVERSION 4—2

4.3. PROGRAM CONVERSION 4—3
4.3.1. Basic Assembly Language Conversion 4—3
4.3.2. FORTRAN Conversion 4—4
4.3.3. COBOL Conversion 4—5
4.3.4. RPG II Conversion 4—6

PART 3. EMULATOR GENERATION

5. INTRODUCTION TO EMULATOR GENERATION

5.1. GENERAL 5—1

5.2. THE EMULATOR AND EMULATED PROGRAMS 5—1

5.3. THE GENERATION PROCESS 5—2

6. DESCRIPTOR CARD PREPARATION

6.1. GENERAL 6—1

6.2. DESCRIPTOR CARD TYPE A 6—4

6.3. DESCRIPTOR CARD TYPE B 6—29

6.4. DESCRIPTOR CARD TYPE C 6—38

6.5. DESCRIPTOR CARD TYPE D 6—41

6.6. DESCRIPTOR CARD TYPE E 6—48

6.7. DESCRIPTOR CARD TYPE F 6—55

7. OPERATING CONSIDERATIONS

7.1. GENERAL 7—1

7.2. DEFINING EMULATOR MAIN STORAGE REQUIREMENTS 7—1

7.3. RUNNING SG$EMJCL UNDER SPOOLING 7—1

7.4. 9200/9300 VSN DUPLICATION FOR 8411/8414 EMULATION 7—2

7.5. RESTRICTIONS FOR DISC PREP 7—2

7.6. RELEASING PERIPHERALS NOT REQUIRED FOR PROGRAM BEING EMULATED 7—2

7.7. SYSTEM HANG BECAUSE OF EMULATOR 7—2

7.8. EMULATOR HANG BECAUSE OF /FINIS CARD 7—2

PART 4. APPENDIXES

A. EMULATION AIDS

A.1. THE 8410 EMULATION AID A—1
A.1.1. Dumping Your 8410 Discs to Tape or Cards A—4
A.1.1.1. Header Card A—4
A.1.1.2. Dump Limit Card A—5
A.1.1.3. END Card A—6
A.1.2. Restoring Your 8410 Disc Image to a 90/30 Disc A—7
A.1.2.1. PIMAGE Control Card Preparation A—7
A.1.2.2. Initializing the PIMAGE Routine A—11
A.1.3. Operational Phases of the PIMAGE Routine A—11
A.1.3.1. Tape Verification — Phase 1 A—11
A.1.3.2. Disc Initialization — Phase 2 A—11
A.1.3.3. Data Restore — Phase 3 A—11
A.1.3.4. Disc Extent Identification — Phase 4 A—12
A.1.4. Restrictions and Considerations in the Use of PIMAGE Routine A—14

A.2. THE 8410 DISC COPYING PROGRAM (COPY$10) A—17
A.2.1. Program Initiation A—17
A.2.1.1. COPY$10 Control Card Preparation A—17
A.2.1.2. COPY$10 Initiation Through Keyin A—20
A.2.2. Program Restrictions A—20

B. EMULATION 9200/9300 OPERATING INSTRUCTIONS

B.1. INITIAL LOADING B—1

B.2. EMULATION LOADING B—2

B.3. SYSTEM CONSOLE SPECIFICATIONS B—2

B.4. EMULATION COMMANDS AND MESSAGES B—3
B.4.1. Program Directives B—4
B.4.2. Display/Alter Commands B—9
B.4.3. Emulator Advisories B—15
B.4.4. Input Message Summary B—15
B.4.4.1. Program Directives B—15
B.4.4.2. Display/Alter Commands B—16

B.5. EMULATOR ERROR CODES B—16

C. CONVERSION UTILITIES

C.1. DISC FILE CONVERSION — SPERRY UNIVAC 9200/9300 TO SPERRY UNIVAC 90/30 VIA OS/3 UNLOAD/RELOAD CONVERSION UTILITY C—1
C.1.1. Disc File UNLOAD Program C—2
C.1.1.1. Program Objectives C—3
C.1.1.2. Interfacing With UNLOAD C—3
C.1.1.3. Input Requirements to UNLOAD C—4
C.1.1.4. Output Files Produced by UNLOAD C—8
C.1.1.4.1. Disc Output C—8
C.1.1.4.2. Tape Output C—9
C.1.1.4.3. Card Output C—12
C.1.1.4.4. Printer Output C—12
C.1.1.5. Preparing the UNLOAD Program for Use C—12
C.1.1.6. Executing UNLOAD C—14
C.1.1.7. Error Processing C—15
C.1.1.8. Hardware Requirements C—18
C.1.2. Disc File RELOAD Program C—18
C.1.2.1. Intermediate Files — Input to RELOAD C—18
C.1.2.1.1. Card File C—19
C.1.2.1.2. Tape File C—19
C.1.2.2. Conversion Mechanism for RELOAD C—19
C.1.2.3. PARAM Card Preparation C—20
C.1.2.4. Job Control Interface to RELOAD C—25
C.1.2.5. Job Control Stream Examples for RELOAD C—27
C.1.2.6. Error Processing Performed by RELOAD C—32
C.1.2.6.1. Error Processing for Intermediate Files C—32
C.1.2.6.2. Error Processing for Output Files C—33
C.1.2.7. Main Storage Requirements C—33
C.1.2.8. Record Format for OS/3 C—34
C.1.2.9. RELOAD Messages C—35

C.2. BASIC ASSEMBLY LANGUAGE TRANSLATION BY SOURCE CODE ANALYZER ROUTINE C—41
C.2.1. Altering Declarative Data Management Calls C—41
C.2.2. Altering Supervisor and Imperative Data Management Calls C—43
C.2.3. Altering Proc Definitions C—44
C.2.4. Direct Address Simulation C—44
C.2.5. Conversion of Privileged and Miscellaneous Instructions C—46
C.2.6. Input to SCAN C—47
C.2.7. Output From SCAN C—49
C.2.7.1. Module Sequencing C—50
C.2.7.2. Specifying Input/Output Information C—50
C.2.8. Related Software Components and Interfaces C—54
C.2.9. Data Base C—54
C.2.10. Error Processing C—54

C.3. CONVERTING YOUR EMULATED 8410 DISC FILES TO OS/3 DATA FILES BY DCON92 PROGRAM C—55
C.3.1. Interfaces Required to Perform File Conversion C—55
C.3.2. Input Requirements to DCON92 C—56
C.3.2.1. DCON Control Card Preparation C—57
C.3.2.2. File Description Card Preparation C—60
C.3.2.3. RELOAD PARAM Card Preparation C—61

C.3.3. Data Bases Required for Conversion C—65
C.3.4. Initiating the DCON92 Program C—65
C.3.5. Console Messages C—66
C.3.6. Example of Input Control Stream for DCON92 C—66
C.3.7. Restrictions in the Use of DCON92 C—66

D. DEVICE CODES

INDEX

USER COMMENT SHEET

FIGURES

1—1. Logical Job Processing 1—4
1—2. Emulation Processing 1—5

3—1. System Transfer 3—5
3—2. Emulation Load Time Considerations 3—6

5—1. Basic Steps in EMGEN Process 5—3
5—2. Emulation 9200/9300 System Description Form 5—4

6—1. Sequence of Cards for Performing EMULAT Phase Functions 6—3

A—1. Overview of 8410 Disc Transfer to 90/30 Disc A—2
A—2. Dump Control Card Sequence A—4
A—3. Multiple-File Configurations A—16
A—4. Alternate Method of Multiple-File Arrangement A—17

C—1. UNLOAD/RELOAD Disc File Conversion Utility C—1
C—2. UNLOAD to SAM File Conversion C—2
C—3. Format of Output Tape Generated by UNLOAD C—10
C—4. UNLOAD Structure During Execution C—14
C—5. Typical Job Control Stream for File Conversion C—15
C—6. Input/Output Interface Requirements for DCON92 Conversion Utility Program C—56
C—7. Input Card Sequence to DCON92 Conversion Utility Program C—56

TABLES

1—1. Emulation Counterparts 1—2

2—1. System Main Storage Requirements 2—1
2—2. Emulation 9200/9300 Main Storage Requirements 2—2
2—3. 9200/9300 Packs per 90/30 Disc 2—3

3—1. 9200/9300 System Support 3—4

4—1. Basic Differences Between 9300 and OS/3 FORTRAN 4—4

6—1. 9200/9300 Descriptor Card Matrix 6—2
6—2. Tape Mode Characteristics 6—45

A—1. Header Card Format A—5
A—2. Dump Limit Card Format A—5
A—3. End Card Format A—7

C—1. File Specification Card Format C—5
C—2. Halt Specification Card Format C—7
C—3. First File Directory Record Format C—10
C—4. Second File Directory Record Format C—11
C—5. Output Card Format C—12
C—6. UNLOAD Error Messages C—16
C—7. Job Control Statements Associated With RELOAD C—27
C—8. RELOAD Warning Messages C—36
C—9. RELOAD Error Messages C—36
C—10. RELOAD Information Messages C—40
C—11. DTF Listing C—43
C—12. File Description Card Format C—60

D—1. Emulator Device Codes D—1

PART 1. EMULATION

1. Introduction to Emulation and Conversion

1.1. GENERAL

The data processing manager's enthusiasm for a larger, more powerful computer must be tempered by a critical evaluation of the compatibility question. He must establish what it will cost, in time and manpower, to convert his current system and devise some way of maintaining at least a limited level of production while undertaking the system conversion. At the same time he must consider the replacement paradox: To justify the turmoil created by change, the newer system must have superior software, superior hardware, or both. More often than not, the newer system's superiority can be measured in terms of its lack of similarity to the existing system. If the differences are major and no solutions to the compatibility issues are offered, the cost of replacement could easily outweigh the advantages.

This manual addresses itself to the compatibility issues and the solutions offered by the SPERRY UNIVAC Operating System/3 (OS/3). To maintain production while converting your SPERRY UNIVAC 9200/9300 System to a SPERRY UNIVAC 90/30 Data Processing System, OS/3 offers programmed emulation, to complement hardware's microcode, and emulation aids to facilitate program and data transfer. This solution is discussed in Part 1. To assist you in the actual conversion effort, OS/3 provides utilities for file transfer and a program to analyze source code disparities. This solution is discussed in Part 2. Until the system has been converted, emulation can maintain production. Part 3 describes how to generate the emulator needed to support emulation. The manual's appendixes detail operational considerations of both the emulation and conversion programs.

First, however, let's review and illustrate the facts of compatibility which are germane to both emulation and conversion, and the degree of consistency between your existing complement of peripherals and those offered on 90/30 systems. Table 1—1 compares the peripherals available on your present system to those of the 90/30.

Table 1—1. Emulation Counterparts

9200/9300 Peripherals	90/30 Peripherals	
	Convert to	Emulate on
0768 printer	0768 printer	Any 90/30 printer
1004 printer BAR printer	0770 printer 0773 printer 0776 printer	Any 90/30 printer or channel-connected BAR printer
0711 reader 0716 reader 1001 reader [1] 1004 reader	0716 reader 0717 reader	Any 90/30 reader [1]
0604 punch	0604 punch	0604 punch
0603 punch 1004 punch	0604 punch 0605 punch	Any 90/30 punch
0920 paper tape	0920 paper tape	0920 paper tape
2703 ODR	2703 ODR	2703 ODR
VI-C tape [2] UNISERVO 12 tape	VI-C tape UNISERVO 12 tape UNISERVO 16 tape UNISERVO 20 tape	Any 90/30 tape or Any 90/30 disc [2]
8411 8414 8410 disc	Not required Not required 8411 disc 8414 disc 8416 disc 8418 disc 8430 disc	8411 8414 8410 is emulated on any disc.
Data communication subsystem	Communication adapter	Not emulated

NOTES:

[1] The 1001 collate feature is not emulated.

[2] Only tape libraries are emulated on disc.

During the conversion interval, virtually all installations are required to maintain some level of production. In many cases there will be programs which will never be converted either because they have a limited life span or because their infrequent utilization does not justify the conversion effort. But, whether or not programs will ultimately be converted, there are some rather apparent ground rules for maintaining the work load:

- Existing programs must run on the SPERRY UNIVAC 90/30 Data Processing System. The cost of maintaining parallel systems cannot be justified.

- Existing programs must run in the environment they were written to comprehend. Any program modifications required by the 90/30 environment merely add to the replacement burden.

- Existing programs, even those with limited use expectancy, may require occasional updates. Provision must be made for this contingency.

- Existing programs cannot be permitted to usurp the 90/30. Allowance must be made for testing and running converted programs.

OS/3, in conjunction with 90/30 microcode, fulfills each of these requirements.

- Your existing programs will run on the 90/30 system. Microcode, working in conjunction with software, permits execution of every instruction in your present system's repertoire.

- You may run your existing system on the 90/30 system, without modification, or you may elect to take advantage of the disc file substitution for card data sets option offered with emulation.

- If problem program updates are required, you may change the existing code, recompile, and link — using the software which has been transferred from your present system to the 90/30 system.

- You may run existing programs under emulation with other emulated programs, or you can run emulated programs with converted programs. However, there must be a separate copy or version of the emulator to control each program being emulated concurrently. Concurrency permits taking fullest advantage of available main storage, plus increasing familiarity with 90/30 system procedures as you move toward full conversion.

1.2. HOW EMULATION WORKS

There are three components of OS/3 emulation: OS/3 emulation aids, the OS/3 emulation program, and the 90/30 hardware microcode.

Emulation aids are software utility programs. They are used to unload your present system, file your tape programs on 90/30 discs, and restore your present system's image on 90/30 discs. The emulation program is a problem program (as is any program you write for or convert to the 90/30 system) which runs under control of the OS/3 supervisor. Your existing programs, files, transactions, supervisor, and job control program are all viewed as data elements by this emulation program.

Hardware microcode executes all instructions — the 90/30 repertoire as well as those of your present system. Microcode is designed to generate a program exception interrupt when it encountered I/O commands, privileged mode instructions, or illegal operation codes. These illegal operation codes may be legal functions (such as storing the processor state control word) which must be viewed as illegal to avoid ambiguity, or actual repertoire exceptions which the emulator has deliberately set to trap tape or disc I/O at the functional level. When such exceptions are noted, microcode causes an interrupt to the OS/3 supervisor, which in turn calls the emulation program for an interpretive execution of the I/O process your program has specified.

Figures 1—1 and 1—2 describe emulation. Figure 1—1 illustrates how the emulator handles jobs; Figure 1—2 is a flowchart of the relationships between the emulator, emulated program, supervisor, and hardware microcode.

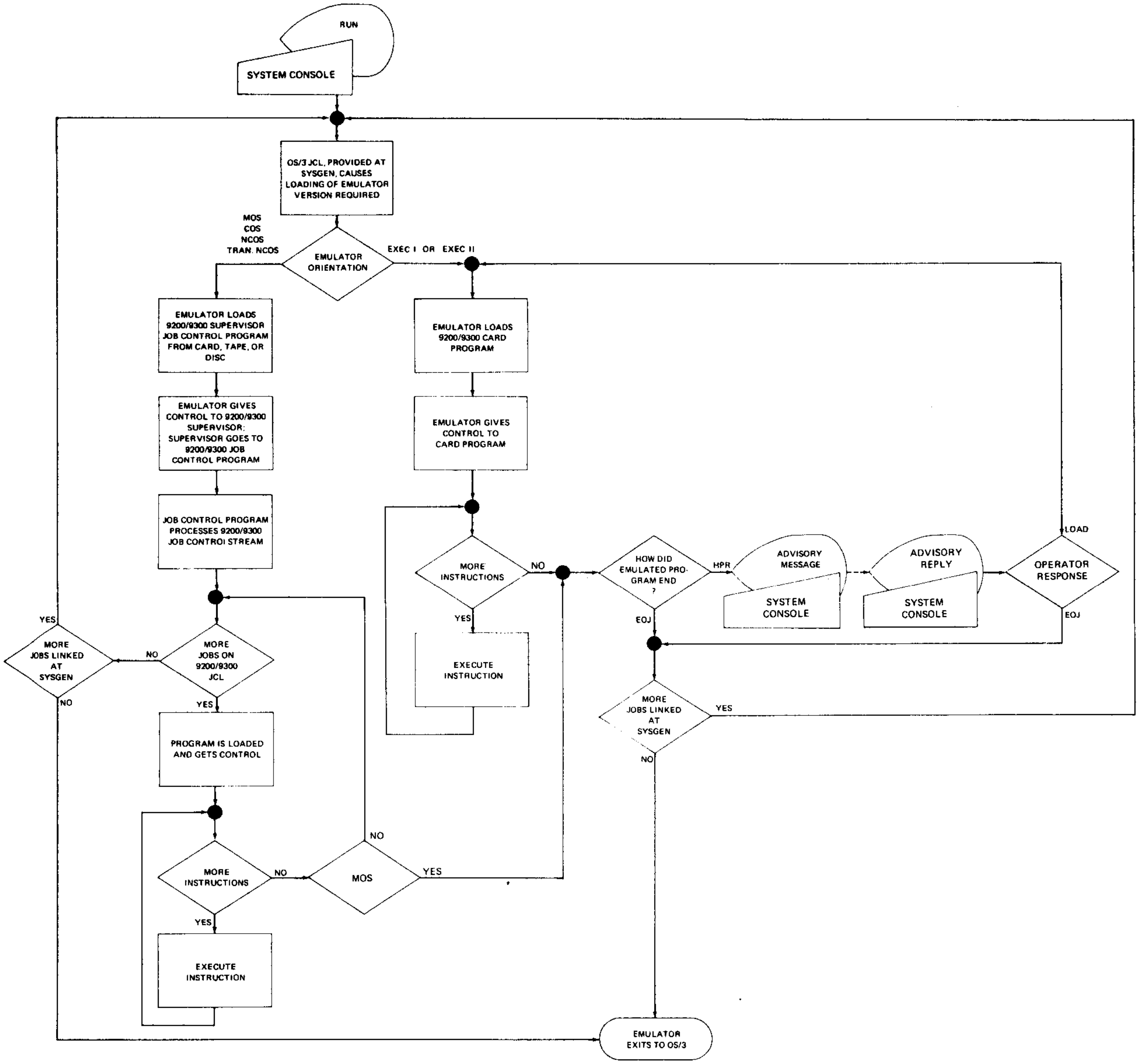

RUN
SYSTEM CONSOLE
OS/3 JCL, PROVIDED AT SYSGEN, CAUSES LOADING OF EMULATOR VERSION REQUIRED
MOS
COS
NCOS
TRAN. NCOS
EMULATOR ORIENTATION
EXEC I OR EXEC II
EMULATOR LOADS 9200/9300 SUPERVISOR JOB CONTROL PROGRAM FROM CARD, TAPE, OR DISC
EMULATOR GIVES CONTROL TO 9200/9300 SUPERVISOR: SUPERVISOR GOES TO 9200/9300 JOB CONTROL PROGRAM
JOB CONTROL PROGRAM PROCESSES 9200/9300 JOB CONTROl STREAM
EMULATOR LOADS 9200/9300 CARD PROGRAM
EMULATOR GIVES CONTROL TO CARD PROGRAM
MORE INSTRUCTIONS
NO
YES
EXECUTE INSTRUCTION
HOW DID EMULATED PROGRAM END ?
HPR
ADVISORY MESSAGE
SYSTEM CONSOLE
ADVISORY REPLY
SYSTEM CONSOLE
OPERATOR RESPONSE
LOAD
EOJ
EOJ
MORE JOBS LINKED AT SYSGEN
YES
NO
YES
MORE JOBS LINKED AT SYSGEN
NO
NO
MORE JOBS ON 9200/9300 JCL
YES
PROGRAM IS LOADED AND GETS CONTROL
MORE INSTRUCTIONS
NO
YES
EXECUTE INSTRUCTION
MOS
NO
YES
EMULATOR EXITS TO OS/3

Figure 1—1. Logical Job Processing

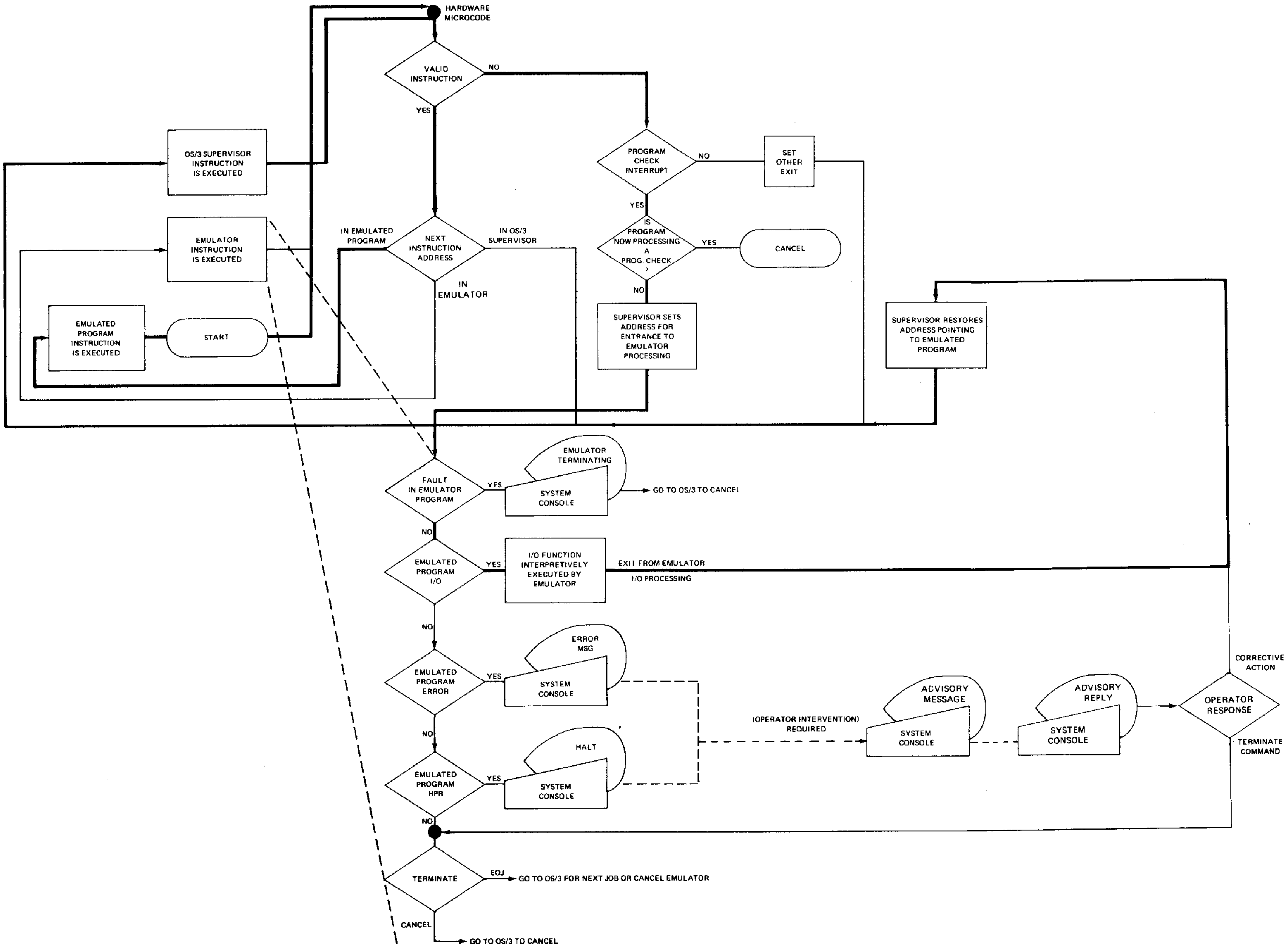


Figure 1—2. Emulation Processing

1.3. WHAT WILL AND WILL NOT BE EMULATED

The emulation capability of OS/3 focuses on batch programs written to operate with standard software. This classification would, in general, include programs written for:

- Card-oriented card systems (EXEC I or EXEC II)
- Card-oriented tape systems (MOS)
- Card-oriented disc systems (MOS)
- Tape-oriented tape systems (NCOS)
- Disc-oriented tape systems (NCOS and transient NCOS)
- Disc-oriented disc systems (NCOS and transient NCOS)

More specifically, this includes SPERRY UNIVAC 9200/9300 EXEC I, EXEC II, minimum operating system (MOS), nonconcurrent operating system (NCOS), transient NCOS, and concurrent operating system (COS). In addition to the problem programs written for standard software, emulation supports those software elements which would be needed to update programs or to manipulate unfamiliar file structures. This category would include:

- BAL assemblers
- RPG compilers
- COBOL compilers
- FORTRAN compilers
- Linkage editors
- Sort/merge programs

Because they are nonstandard, user-written supervisors, and programs which depend upon those supervisors, are excluded. At the same time, emulation cannot support programs or functions which depend upon the hardware characteristics of your current system, such as peripheral prep routines; low-order main storage protection (other than that available with the 90/30), or console-entered stops. Instruction stepping is permitted, because the 90/30 maintenance panel includes this capability. When stepping instructions of an emulated program, however, the observer will frequently encounter instructions within the emulator itself. The less obvious exclusions are:

- Console inquiry
- Checkpoint-restart

- 9200/9300 symbionts
- Card-oriented macro pass and compressor
- Multiply and divide packed exceptions:
  - If the length of operand 2 is equal to or greater than that of operand 1, or
  - If the multipler, operand 2, exceeds 15 digits or 8 bytes, or
  - If either operand contains invalid digits or signs.

The emulation capability does not support the 9200/9300 disc prep routine due to the fact that the routine is not written to operate under the standard 9200/9300 software. The discs may be prepped on either the 9200/9300 or on the OS/3 and then used under emulation. The discs, however, cannot be prepped under emulation.

Emulation also cannot support any program which utilizes a main storage wrap-around condition because the emulated program simply indexes through the 90/30 main storage.

2. Planning for Emulation

2.1. GENERAL CONSIDERATIONS PRIOR TO EMULATION

First, of course, you must decide which programs are to be emulated and check to make sure that they contain nothing which would preclude their emulation. Then a SPERRY UNIVAC 90/30 configuration must be specified which includes functionally equivalent peripherals and sufficient main storage (see 2.1.3). The main storage requirements for emulation are listed in Table 2—1.

Table 2—1. System Main Storage Requirements

9200/9300 System to be Emulated		90/30 System Required for Emulation	
Type	**Size**	**Minimum**	**Maximum**
Card	8K,12K, 16K, or 24K	32K	49K
Card	32K	49K	65K
Disc	12K or 16K	32K	65K
Disc	24K	49K	65K
Disc	32K	49K or 65K	65K
Disc (with tape or other special peripherals)	As required	49K	65K
Tape	8K, 12K, 16K, 24K, or 32K	32K	65K

The main storage requirements for the emulation program itself are conditioned by the configuration you define when tailoring the emulator. The ranges are shown in Table 2—2.

Table 2—2. Emulation 9200/9300 Main Storage Requirements

Emulator Main Storage Requirements	Bytes
Basic unit record handler	3000 – 5500
Basic disc handler*	4000
Tape handler*	2500
Expanded, mux channel, unit record handler*	1000 – 4000
Total – emulator Add 9200/9300 main storage Main storage for emulation	xxxxx xxxxx **

* Included only when the peripheral is specified at system generation.
** Basic example – user should calculate own specific storage requirements.

Once the SPERRY UNIVAC 90/30 system configuration has been established, using Table 1—1 for peripheral selection and Tables 2—1 and 2—2 for main storage estimation, you must plan the transfer of your present system's image. The process involves:

- Transporting data files
- Transferring programs
- Creating one or more emulator programs
- Specifying device utilization
- Equating print line controls
- Generating 90/30 job control streams

2.1.1. Data Files

There are emulation aids which dump discs and restore data to SPERRY UNIVAC 90/30 discs. In Part 2 and Appendix C you'll find that there are conversion utilities which dump discs and restore data. Although they appear to be the same, they are functionally quite different. An understanding of this difference is crucial to an understanding of emulation's scope. The emulation aids extract an image of your entire system, without regard to content, and re-create a bit-for-bit copy on 90/30 discs. The conversion utilities are concerned with data files and their reconstruction as files on the 90/30. Emulation aids, in other words, are disc *pack* oriented, while conversion utilities deal with *files* or *records*. The emulation aid, which is used to dump the 8410 disc, is detailed in Appendix A. This appendix also contains a description of the emulation aid which is used to restore your system's image on the 90/30 disc you select.

The 90/30 discs, which you specify for emulation, will have an equal or greater capacity than those in your present system. Table 2—3 delineates that relationship. Care should be taken to avoid overpacking these 90/30 discs. Although you can, at least in theory, get by with fewer 90/30 disc drives, the programs which are to be emulated may presume more online disc packs than you have drives available. This is the same problem you encountered when assigning multiple files to your present packs.

The 8411 and 8414 discs present a somewhat exceptional case. Because these drives are physically supported on the 90/30, no provision is necessary for dumping 9200/9300 series-written packs and restoring them to any 90/30 disc. If you now have either the 8411 or 8414 drives, and require their packs to support emulation of your 9200/9300 series programs, you must have equivalent disc drives on your 90/30. These packs may be used to store programs, system elements, data files, or all three. The data files may be written with any of the standard, 9200/9300 series access methods. Although the emulator normally employs OS/3 data management for emulated files, 8411 and 8414 files, because of the special circumstances involved, are dealt with on the physical I/O level.

If your system is now card or tape oriented, it is transportable to the 90/30 without prior manipulation. Once there, you may elect to take advantage of the disc filing option prior to emulation of your 9200/9300 series programs (see 3.1).

Table 2—3. 9200/9300 Packs per 90/30 Disc

9200/9300 Disc	Number of Packs Which May Be Stored on 90/30 Discs				
	8411	8414	8416	8418	8430
8410 (one side)	2	7	10	10 low density 22 high density	21
8410 (two sides)	1	3	5	5 low density 11 high density	10

NOTE:

No allowance is included for OS/3 storage requirements.

2.1.2. Transferring Programs

Emulation makes no distinction between data files and program files; it is insensitive to structure and content. If your programs are now disc resident, they are removed by the same emulation aid which dumps your files and system software. If they are card or tape resident, they are transportable in their present form. OS/3 offers the choice between leaving card programs in their present form or storing them in the OS/3 system JCS library file ($Y$JCS). You may decide to leave your programs on tape, or create a disc file. The emulation aid which creates (and catalogs) this program file is also described in Appendix A.

2.1.3. Creating Emulators

The last four steps in the transfer process — creating emulators, specifying devices, equating print line controls, and generating job streams — are all built into the OS/3 system generation process and are more fully described in Part 3. OS/3 system generation is limited to the specification of a supervisor, an emulator, and a communications network. Emulator specification is a self-contained subset of the system generation process and can be utilized as many times as your system demands.

An emulator reflects the hardware characteristics and supervisor orientation of your present system. An emulator is generated to support a specific main storage size and peripheral complement to recognize the location of your current system, software, job control, and disc data files.

3. System Preparation

3.1. PREPARATORY STEPS

The SPERRY UNIVAC 90/30 Data Processing System is a disc-based computer system. If your previous system was entirely disc oriented and if your supervisor, job control program, data files, and problem programs were all disc resident, the transition will be smoother and your emulated system will run faster than one which is card based.

If your earlier system was not entirely disc based, there are preparatory steps you could take which, in most cases, would result in increased throughput. Each of these steps involves transferring part of your present system to disc before emulating. You can follow each applicable step, or you can ignore any which do not suit your applications. Your programs will still be emulated. The choice between storing portions of your existing system on disc or running them in their present form is one of expedience. If you are going to emulate only a few programs, the programs are small or infrequently used, or the data files are limited, disc storage may not be worth even the minimal effort involved.

Just how you go about emulating your current system depends largely upon that system's orientation and the extent to which you modify that orientation before execution on the 90/30. As noted earlier, if you are now entirely disc based, you need no further preparation. You may skip the balance of this discussion and go to Appendix B, which each site is provided space for determining the preferred mode of operation.

3.2. METHODS OF ACCESS

You have two ways of accessing the 90/30 system and three approaches to emulation. You gain access by using emulation aids or by putting cards in the system reader — or both. You execute from disc, tape, card, or a combination of the three. Emulation aids, detailed in Appendix A, are used to transfer the image of the 8410 disc packs to the SPERRY UNIVAC 90/30 disc subsystem you specify or to catalog your tape programs on these same discs. The emulation aids, unless you are totally card oriented, are used with your present system before you generate one or more versions of the emulator program.

The system reader feeds load modules and job control language (JCL) to OS/3. They may be fed once and filed in the OS/3 system JCS library file ($Y$JCS), or fed each time you wish to emulate a program. In either case, they must be fed at least once.

The first deck of cards placed in the system reader is always the OS/3 job control stream which was produced as a by-product of system generation. With the JCL deck you will get a listing; two cards will be noted. If you wish to run using the OS/3 temporary job run library file ($Y$RUN), you insert your current system's cards between the two OS/3 JCL statements marked. If you are EXEC I or EXEC II oriented, you insert one or more card programs; if you run with a card supervsior, you precede your programs, just as you are now doing, with the supervisor, job control program (if used), and the 9200/9300 job control statements. You then issue a FILE command at the 90/30 system console; OS/3 automatically goes to the system reader (see Appendix B).

On the other hand, you may wish to retain your current mode of operation — running specific card decks through the system reader each time a program is to be emulated. In this case, you still begin with the OS/3 JCL deck but, instead of inserting your current system, you simply add each element, as you are now doing, behind this JCL deck.

These basic approaches to emulating — running everything from OS/3 $Y$JCS or executing each job as a separate system reader entry — are, of course, for the card-oriented users. They will not answer all the possibilities faced by minimum operating system (MOS), concurrent operating system (COS), nonconcurrent operating system (NCOS), or transient NCOS users, and even card-oriented users will achieve greater efficiencies by employing variations of these basic approaches. To understand how these variations may be implemented requires a closer look at the structure of the emulator itself.

3.3. SYSTEM GENERATION

System generation tailors an emulator program to correspond with your current hardware configuration and software orientation. The hardware configuration portion tells the emulator which 90/30 peripheral to substitute when, for example, your program calls upon the bar printer. The software or system specification gears the emulator to look for your current system elements on disc, on tape, or from cards. If you now have a disc system, you parameterize a disc-oriented emulator and that emulator expects to find your supervisor on disc. It won't expect to find it on cards. Conversely, if you say that you are card oriented, the emulator will not look for your system elements on tape. Those installations which might conceivably alternate between card-oriented and disc-oriented programs, in order to achieve greater utilization from a limited number of disc drives, would have to generate a version of the emulator for both card and disc systems — and ensure that the proper emulator was associated at run time. The disc-oriented emulator would run with the disc-oriented programs; programs, written for card-resident software, would be emulated by the separate, card-resident emulator. EXEC I and EXEC II programs are the single exception to this one-for-one equation of operating systems and emulators. Card programs may be accommodated in either of two ways: They may be embedded in the previous system's job control stream or they may be separately loaded and executed by a degraded disc, tape, or card-resident emulator. If they are embedded, the emulator treats them as just so many card images which are, presumably, intelligible to the system being emulated. Card programs are separately loaded using the LOAD command. Upon receipt of this command, the emulator dynamically reconfigures itself and proceeds to execute card programs. This reconfiguration is the functional equivalent of rebooting your previous system to overlay the resident software with EXEC I or EXEC II. Once emulator operation has been degraded, a new version must be loaded to handle card-resident, disc, or tape programs (jobs).

The term *job* requires further clarification because OS/3 emulation adds a dimension to this concept. Each EXEC I or EXEC II program is identified as a job. With a 9200/9300 job control stream, you mak link several executions under one job. This concept remains unchanged. EXEC I and EXEC II programs must be serially loaded and executed. Systems employing job control streams can still link any number of executions into a single job. OS/3 emulation permits linking from one job to the next — whether the jobs are EXEC I or EXEC II programs or JCL-driven control streams. This linkage is specified as one of the system generation parameters and implemented by the control cards which are automatically produced by the generation process. At first, you will probably restrict linking to logically related 9200/9300 programs. Later, as you become more familiar with OS/3 job control statements, you will add OS/3 elements, such as the file utility or sort/merge, to the chain of jobs to be executed.

The load time variations which may be employed are conditioned by the version of the emulator you are using (see Figure 3—2) by the job sequencing you wish to implement.

3.3.1. File OS/3 JCL Only

The job control stream provided as part of system generation may be stored in OS/3 $Y$JCS. At run time, under these conditions, your system reader would contain:

- EXEC I or EXEC II card programs and, if required, card data files; or

- Card-oriented supervisor, followed by:

  - the job control program from the 9200/9300 (if needed),

  - the 9200/9300 JCL cards,

  - card programs,

  - card data files (if required); or

- 9200/9300 JCL cards followed by card programs and then card data (if required).

3.3.2. File OS/3 JCL and System Elements

The effect of this approach is to put card-oriented systems on an efficiency level with disc and, to an extent, tape systems. Because there are no system elements involved in running card programs and no cards are fed from the reader to disc or tape systems, the effect of this approach is merely to reduce the number of cards for the card-oriented system user.

The emulation process is initiated by entering a RUN command at the 90/30 system console — with job name if the OS/3 JCL is in $Y$JCS, or without job name for system reader input. The emulator knows, by virtue of its system generation tailoring, whether a supervisor and job control program are required and, if so, whether to find these elements on card, tape or disc.

If the emulator's structure indicates that no system elements are required, the emulator will assume that an EXEC I or EXEC II program has been filed on disc with the OS/3 JCL. If the specified program cannot be found on disc, input is expected to be in the system reader. The card program, once loaded, runs until it attempts either an end-of-job (EOJ) or HPR. If the EOJ occurs, the emulator returns to OS/3 job control to load a subsequent card program — with the same or different version of the emulator program. When the emulated program does an HPR, the emulator displays a HALT message and awaits a response. The operator is, of course, expected to know whether the HPR condition displayed warrants job termination or the execution of a subsequent card program. If another card program is to be run, the operator keys in a LOAD command (meaning that the card program is in the system reader) or a LOAD DISC (the card program has been filed).

When the emulator's structure indicates that 9200/9300 system elements are involved, those system elements are loaded during emulator initialization. Card-oriented elements are assumed to be on disc with the OS/3 JCL. They are loaded from the system reader only when they cannot be located on disc. The 9200/9300 system elements, running under emulation, direct the loading and execution of those jobs identified in the 9200/9300 job control stream. System controlled-emulation can be terminated to permit loading of an EXEC I or EXEC II card program. Normally, this would be done when the emulator stops to display an emulated program's HPR message, or when the emulator is forced into a HALT condition by an error in the emulated program. Here again, the operator uses the LOAD or LOAD DISC command. Once degraded by the running of a card program, the emulator will only handle additional EXEC I or EXEC II programs. A new job initiation is required to return to card-oriented, disc-oriented, or tape-oriented software.

Figure 3—1 and Table 3—1 illustrate the steps which could be taken before executing programs under emulation. Figure 3—2 displays the load time considerations.

Table 3—1. 9200/9300 System Support

System Element	EXEC I or EXEC II	MOS	Tape NCOS Tape COS	Disc NCOS Transient NCOS, COS
Card programs	Yes	Yes	Yes	Yes
Card data	Yes	Yes	Yes	Yes
Tape programs	No	No	Yes	No
Tape data	No	Yes	Yes	Yes
Disc programs	No	Yes	No	Yes
Disc data	No	Yes	Yes	Yes

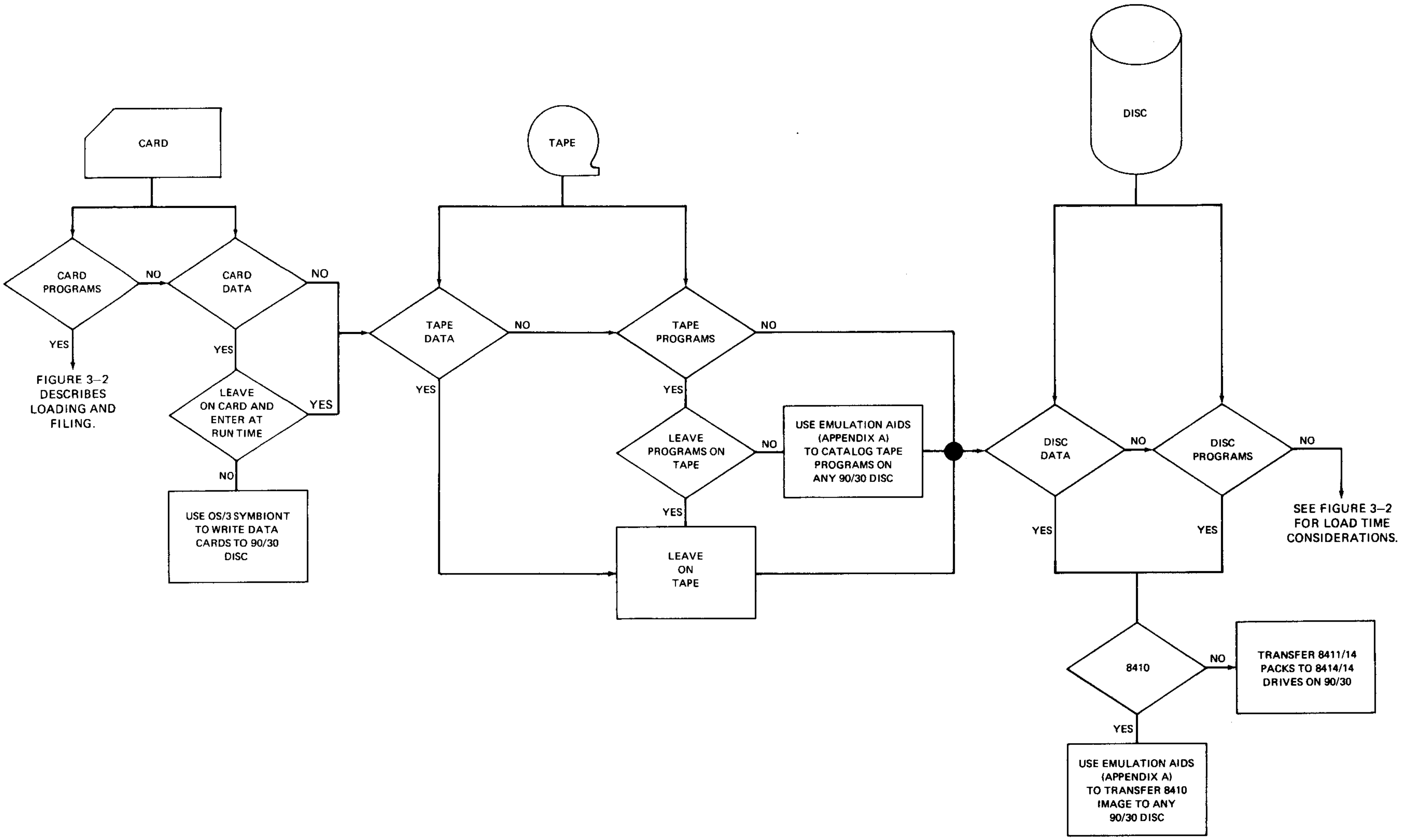

CARD
CARD PROGRAMS
NO
YES
FIGURE 3–2 DESCRIBES LOADING AND FILING.
CARD DATA
NO
YES
LEAVE ON CARD AND ENTER AT RUN TIME
YES
NO
USE OS/3 SYMBIONT TO WRITE DATA CARDS TO 90/30 DISC
TAPE
TAPE DATA
NO
YES
TAPE PROGRAMS
NO
YES
LEAVE PROGRAMS ON TAPE
NO
USE EMULATION AIDS (APPENDIX A) TO CATALOG TAPE PROGRAMS ON ANY 90/30 DISC
YES
LEAVE ON TAPE
DISC
DISC DATA
NO
YES
DISC PROGRAMS
NO
YES
SEE FIGURE 3–2 FOR LOAD TIME CONSIDERATIONS.
8410
NO
TRANSFER 8411/14 PACKS TO 8414/14 DRIVES ON 90/30
YES
USE EMULATION AIDS (APPENDIX A) TO TRANSFER 8410 IMAGE TO ANY 90/30 DISC

Figure 3—1. System Transfer

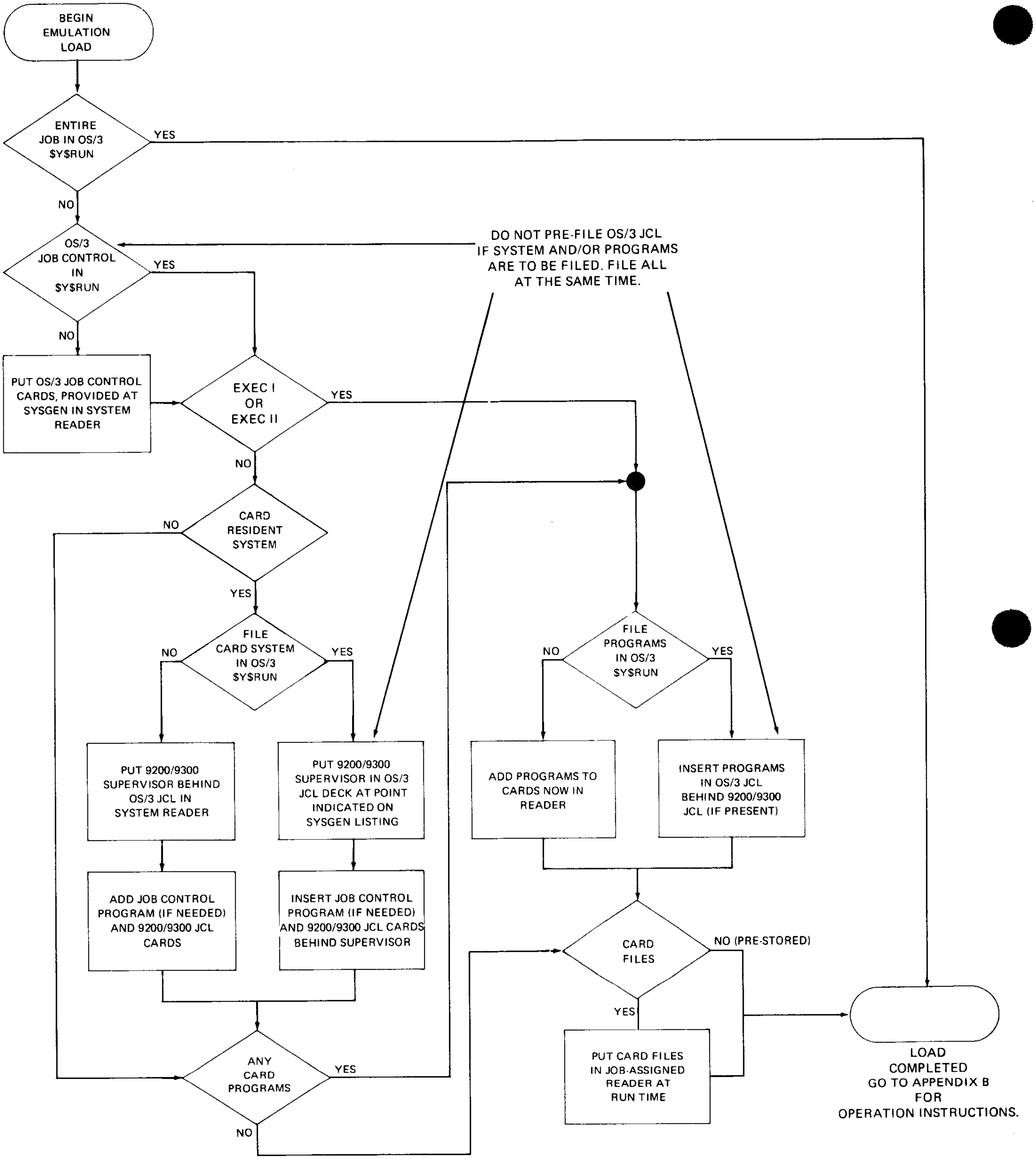

BEGIN EMULATION LOAD
ENTIRE JOB IN OS/3 $Y$RUN
YES
NO
OS/3 JOB CONTROL IN $Y$RUN
YES
NO
DO NOT PRE-FILE OS/3 JCL IF SYSTEM AND/OR PROGRAMS ARE TO BE FILED. FILE ALL AT THE SAME TIME.
PUT OS/3 JOB CONTROL CARDS, PROVIDED AT SYSGEN IN SYSTEM READER
EXEC I OR EXEC II
YES
NO
CARD RESIDENT SYSTEM
NO
YES
FILE CARD SYSTEM IN OS/3 $Y$RUN
NO
YES
PUT 9200/9300 SUPERVISOR BEHIND OS/3 JCL IN SYSTEM READER
PUT 9200/9300 SUPERVISOR IN OS/3 JCL DECK AT POINT INDICATED ON SYSGEN LISTING
ADD JOB CONTROL PROGRAM (IF NEEDED) AND 9200/9300 JCL CARDS
INSERT JOB CONTROL PROGRAM (IF NEEDED) AND 9200/9300 JCL CARDS BEHIND SUPERVISOR
ANY CARD PROGRAMS
YES
NO
FILE PROGRAMS IN OS/3 $Y$RUN
NO
YES
ADD PROGRAMS TO CARDS NOW IN READER
INSERT PROGRAMS IN OS/3 JCL BEHIND 9200/9300 JCL (IF PRESENT)
CARD FILES
NO (PRE-STORED)
YES
PUT CARD FILES IN JOB-ASSIGNED READER AT RUN TIME
LOAD COMPLETED GO TO APPENDIX B FOR OPERATION INSTRUCTIONS.

Figure 3—2. Emulation Load Time Considerations

PART 2. CONVERSION

4. File and Program Conversion

4.1. GENERAL

In order to realize the maximum potential of your 90/30 system, you must become familiar with each OS/3 component and determine the extent to which it can be applied to your task. What must be learned is conditioned by what is now familiar. If, for example, you now run with COS, you are at least acquainted with some of the functions performed by the supervisor, job control program, and job control language. On the other hand, EXEC I or EXEC II users might well have to acclimate themselves to the system environment before they can appreciate the scope and flexibility of the individual components of OS/3.

It is normally impossible to indulge in an extended familiarization interval before putting a new system on a production basis. One way of buying the necessary time is to emulate some or all of your system during the transition phase. Another is to define a learning schedule based upon the immediacy inherent in understanding the fine points of OS/3 elements. Perhaps a combination of the two is appropriate.

The critical areas are data base definition and programming languages. Hasty or ill-advised commitments in these areas will be far more costly than generation of a supervisor which is actually larger than necessary or preparation of a job control stream which fails to release unused peripherals between job steps. On the other end of the scale are elements which may never be used at all, such as the clear disc or tape prep utilities, or components such as communication, which are not scheduled for implementation before the batch system becomes stabilized. Between these extremes are those elements whose requirements must be met, even though an appreciation of the element's treatment of those requirements can be delayed indefinitely. In this category would be OS/3 job control language, linkage-editor parameters, and the keywords required to drive the sort/merge or data utility.

The balance of this section deals with the critical areas of data files and programs, either because upward compatibility exists or because conversion assistance is provided. Both these and other OS/3 components are more completely defined in separate manuals.

For additional information concerning the details of job control, job stream, and system generation, reference should be made to OS/3 documentation. The documents that are of primary importance to the user are:

System messages programmer/operator reference, UP-8076 (current version)

Operations handbook for operators, UP-8072 (current version)

Supervisor user guide, UP-8075 (current version)

Job control user guide, UP-8065 (current version)

Systems service programs user guide, UP-8062 (current version)

Assembler user guide, UP-8061 (current version)

Data utilities user guide/programmer reference, UP-8069 (current version)

Data management user guide, UP-8068 (current version)

Processor programmer reference, UP-8052 (current version)

System description, UP-8060 (current version)

Report program generator (RPG II) user guide, UP-8067 (current version)

Basic COBOL supplementary reference, UP-8057 (current version)

Extended COBOL supplementary reference, UP-8059 (current version)

4.2. FILE CONVERSION

Your data files must be transferred to the 90/30 system. Transfer demands a transportable medium. If your data files are now on tape, they need no further manipulation to permit transfer. Card is also an acceptable medium, although the physical aspects of transfer can be imposing. Your 8410 disc pack is not an acceptable medium because its contents cannot be read on 90/30 disc drives.

OS/3 provides conversion utilities to facilitate transfer of files not on the 8410. Unlike the emulation aids described in Part I and Appendix A, the conversion utilities are sensitive to the file structure, if not content. The emulation aids provide a bit-for-bit transfer, without regard to either structure or content.

The UNLOAD program accepts input from the 8410 disc and writes your files to either tape or card. RELOAD, the counterpart of UNLOAD, runs on the 90/30 and recreates your data files in OS/3 format on any of the disc subsystems supported. While restoring your files, you may alter the file name or modify its structure. If you are satisfied with its present organization, you may accept the RELOAD default values and achieve a one-for-one replacement of your current data base. The UNLOAD and RELOAD conversion utilities are detailed in Appendix C.

Employing your data files with emulated programs does not preclude conversion, even though your 9200/9300 has been released. In this case, the conversion utility, UNLOAD, and the disc system to which it is linked are processed as an emulated program on the 90/30. Because your files were originally transferred by the emulation aids, they now appear to the conversion utility just as they would if they still resided on an 8410. The 8410 image is read from one 90/30 disc and written by UNLOAD to an OS/3 temporary card file. RELOAD is then executed. It reads card images from the temporary file and writes the recreated file to any 90/30 disc. While it is possible to go from an emulated file to a converted equivalent, it is not possible to go the other way or to share a file between an emulated program and one which runs under OS/3.

Your 9200/9300 series system may be equipped with 8411 or 8414 discs, in lieu of the 8410. How these discs are converted depends upon your 90/30 configuration. If you will not have equivalent 8411 or 8414 drives on the 90/30, you would follow the same path which has been described for the 8410 user: Your 8411 or 8414 would be dumped to tape or card on the 9200/9300 series and transported to the 90/30. If your 90/30 will have the same drives, your 8411 or 8414 pack can be used as the transportable medium. In this case you would run the UNLOAD/RELOAD sequence as described for emulated files. Whether or not you first used the files under emulation is, of course, irrelevant.

4.3. PROGRAM CONVERSION

Expediency, as the basis for program conversion, is as tenuous an argument as it is in most situations. If you simply recompile, using the OS/3 version of a now-familiar language, your direction may be more lateral than forward. OS/3 provides six compilers: basic assembly language (BAL), basic FORTRAN, FORTRAN IV, basic COBOL, extended COBOL, and RPG II. Each deserves consideration when you are using emulation programs.

4.3.1. Basic Assembly Language Conversion

The 9200/9300 BAL is a subset of the 90/30 instruction repertoire. Your conversion problems, if any, will not be on the instruction level. Macros and file defintiions (DTFs) will require more of your time. To expedite the conversion of BAL programs, there is a conversion utility SCAN. SCAN runs under OS/3 on the 90/30. Your 9200/9300 source program from card, tape, or disc provides input to SCAN. The program flags system macros and modifies DTFs to correspond with the requirements of OS/3 data management. A source listing with appropriate diagnostics is one type of output from SCAN. All modifications are noted in this listing and special indications are provided for areas within your 9200/9300 program which cannot be interpreted by SCAN. The other type of output is a revised source program output to disc. You specify whichever form of output you desire. SCAN is detailed in Appendix C.

4.3.2. FORTRAN Conversion

OS/3 provides two FORTRAN compilers. The first, referred to as the basic FORTRAN compiler, provides all of the 9300 FORTRAN functions and includes many additional features. FORTRAN IV, the second OS/3 compiler, is an expansion — a superset — of the basic FORTRAN compiler. The object program generated by these OS/3 compilers and executed on the 90/30 will, in terms of processor time, run 30 times faster than the same program compiled and executed on the 9300. If I/O is confined to tapes or discs, total run time will be substantially faster than 9300. Both the basic FORTRAN compiler and FORTRAN IV require the inclusion of the 90/30 hardware floating point feature. The basic FORTRAN compiler runs in a UNIVAC 90/30 with a minimum of 32K storage; the FORTRAN IV compiler requires 65K.

The 9300 FORTRAN numerical programs will be somewhat easier to convert than those which perform extensive character manipulation, primarily due to the differences in internal representation of values. The basic differences between the 9300 and the OS/3 compilers are shown in Table 4—1.

Table 4—1. Basic Differences between 9300 and OS/3 FORTRAN

	9300	OS/3
Integer data type	2-byte, binary; hardware detected overflow	2- or 4-byte binary; hardware detected overflow on 4-byte expressions only
Real data type	6 bytes; decimal-based arithmetic by software 9-digit accuracy Magnitude of $10^{\pm 50}$	4 or 8 bytes; binary arithmetic by hardware 7- or 16-digit accuracy Magnitude of $10^{\pm 75}$
Computed GOTO	Marked by diagnostic index if out of range	Next statement processed if index is out of range
END statement	May have a label	May not have a label
Sin/cos	Accepts any argument	Argument must be $\leqslant$8.2E5
Carriage control	Printer channels 2–6 supported	Carriage control is not supported.
Common/equivalence	May have to be reorganized due to word size differences	
Data tapes	Acceptable if formatted and if they contain no variables written in a format code	
Format statements	May require recoding due to the changes in word size and accuracy	

4.3.3. COBOL Conversion

OS/3 also provides two COBOL compilers, a basic COBOL and an extended COBOL. Both conform to the 1968 American National Institute standard and both are upward compatible with the 9300 compiler. The basic COBOL will run in a 32K 90/30 system; the extended COBOL compiler requires a 48K system.

Although you will probably wish to consider utilization of the OS/3 COBOL added features, your present programs will be compiled if the following exceptions are noted and the implied changes made:

- SYSCHAN is NEXT PAGE, if used, must be changed to a SYSCHAN-t clause. The numeric value for t is found in the COBOL reference manual; for a complete description of the various printer subsystems, see the OS/3 data management reference manual, UP-8069 (current version).

- Label value clauses for standard labels must be removed. Standard labels will be checked by OS/3.

- On the 9300, the ACCEPT identifier presumes control panel entries. OS/3 compilers will reference the job control stream for the source of the ACCEPT statement.

- On the 9300 system, a DISPLAY identifier always directs output to the line printer. With no option specified, OS/3 COBOL will output to the 90/30 system console. If the printer is desired, the *upon* option must be included, and SYSLST defined in the special names paragraph.

- Under the 9300 compiler, programs did not require an INVALID KEY clause on WRITE, if the file had been opened for I/O. OS/3 COBOL will require the INVALID KEY clause.

- The 9300 compiler permits a non-COBOL I/O technique: an INDEX AREA clause as part of the SELECT entry. Under OS/3 COBOL, the APPLY clause must be substituted for INDEX AREA.

- The 9300 compiler allows files to be opened for OUTPUT when ACCESS is RANDOM for INSERTs. OS/3 COBOL does not permit OPEN OUTPUT when ACCESS IS RANDOM. A file must be created sequentially. Once created, it may be randomly accessed.

- In response to a WRITE dataname directive, the 9300 printer would display the specified line and then advance. The American National Institute standard and OS/3 COBOL default option is to do a line advance before printing. To permit either specification, OS/3 COBOL permits a modifier, as:

  WRITE dataname $\left\{ \begin{array}{c} \text{BEFORE} \\ \text{AFTER} \end{array} \right\}$

  If neither option is used, the default is AFTER.

4.3.4. RPG II Conversion

OS/3 RPG II is compatible with programs written for the 9200/9300 compiler, except where peripheral or system incompatibility makes code generation impossible to either duplicate or simulate. In order to minimize the conversion effort, the OS/3 RPG II compiler may be directed to operate in either the 9200/9300 or 90/30 mode. The former mode accommodates all 9200/9300 statements, except those itemized below. The 90/30 mode offers extended capability and is, for that reason, recommended for all new programs. See the referenced RPG II manual for operating details and information concerning expanded capabilities.

The differences between your present compiler and the OS/3 RPG II are:

- Character file names may be no larger than seven characters.
- Multivolume logical unit numbers are not permitted.
- Overflow indicators for output lines are only processed at the overflow time specified in the RPG II cycle.

PART 3. EMULATOR GENERATION

5. Introduction to Emulator Generation

5.1. GENERAL

Emulation of your existing programs involves three steps:

1. Preparing your system for emulation
2. Describing your previous system to SPERRY UNIVAC Operating System/3 (OS/3)
3. Executing your programs under emulation

Normally, you would first prepare for emulation by transferring the image of your previous system to the 90/30 system; then you would generate the emulator or emulators required; and, finally, you would execute your previous system's programs with the appropriate emulator. While there is a chronology involved, there is no specific time frame. You may transfer your files before or at the same time you generate your emulator. You may run your emulated programs immediately after system generation or wait until a more convenient time.

5.2. THE EMULATOR AND EMULATED PROGRAMS

The emulator is a composite model that reflects both the hardware configuration and software orientation of your previous system. It is, in other words, a model of the hardware/software package delivered to you by the manufacturer. The number of emulators you need may be equated to the number of hardware or software configurations you employed in your previous system.

Most users had one computer in which all of the problem programs ran under the same software system. In this case, you would define that hardware configuration and stipulate the software system to the emulator generation process. You would need only one emulator. That emulator would be used to control the problem program to be emulated. The emulator is not, however, shared in a concurrent environment. A separate version of this same emulator is loaded for each program to be executed concurrently.

Two conditions would require generation of more than one emulator. They are:

1. Separate Computers

   If your previous system contained more than one computer, you may require an emulator for each. This possibility could be avoided by generating an emulator that covered both configurations — as long as the same software system was used in both. For example, assume these configurations:

Configuration 1	Configuration 2
32K main storage	16K main storage
1 printer	1 printer
1 reader	1 reader
— — — — — — — — — — —	1 punch
2 discs	— — — — — — — — — — —
1 tape	— — — — — — — — — — —

   You could define unique emulators to control the problem programs that ran in each of these configurations or you could generate one emulator which assumed:

   Composite Configuration

   32K main storage
   1 printer
   1 reader
   1 punch
   2 discs
   1 tape

2. Different Software

   Your previous hardware configuration may have had a limited number of disc drives. Ordinarily, you ran under the disc operating system. At times, however, you ran card programs, or card-resident software, in order to achieve more efficient disc drive utilization. If you wish to emulate both the disc-oriented programs and the card-oriented programs, you will require an emulator for each. You need two emulators for the same reason you were required to reboot your previous configuration — the software, in this case the emulator, must be told where to find software elements and how to load your program.

5.3. THE GENERATION PROCESS

The generation process for creating an emulator, building software images of printer control loops, and producing the OS/3 job control deck needed to execute your program is basically an easy one. That is, you are responsible only for providing the descriptor card input (Section 6) to EMGEN and for initiating the routines of the emulator generation (EMULAT) phase of OS/3 SYSGEN to build your emulator and to produce the OS/3 job control deck. Figure 5—1 illustrates the basic steps in the generation process.

To prepare your descriptor cards, it is suggested that you first describe your emulation/emulator requirements by preparing the emulation 9200/9300 system description form UD1-1220 (Figure 5—2).

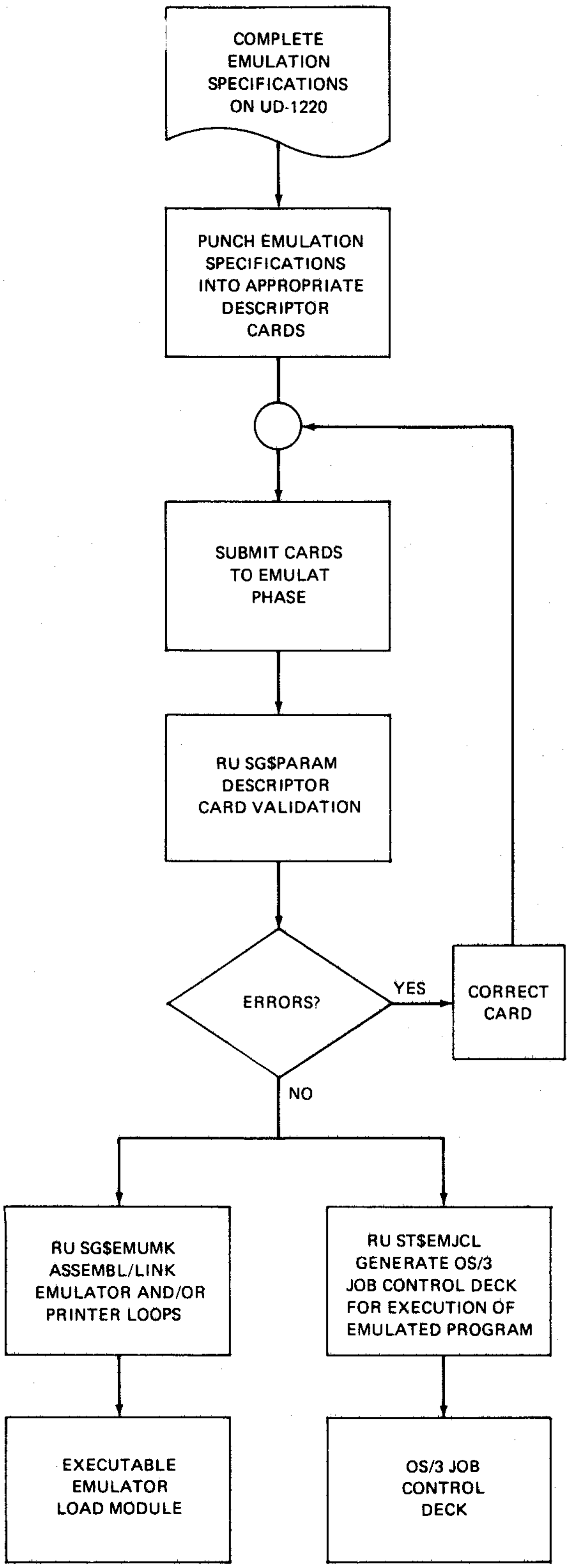
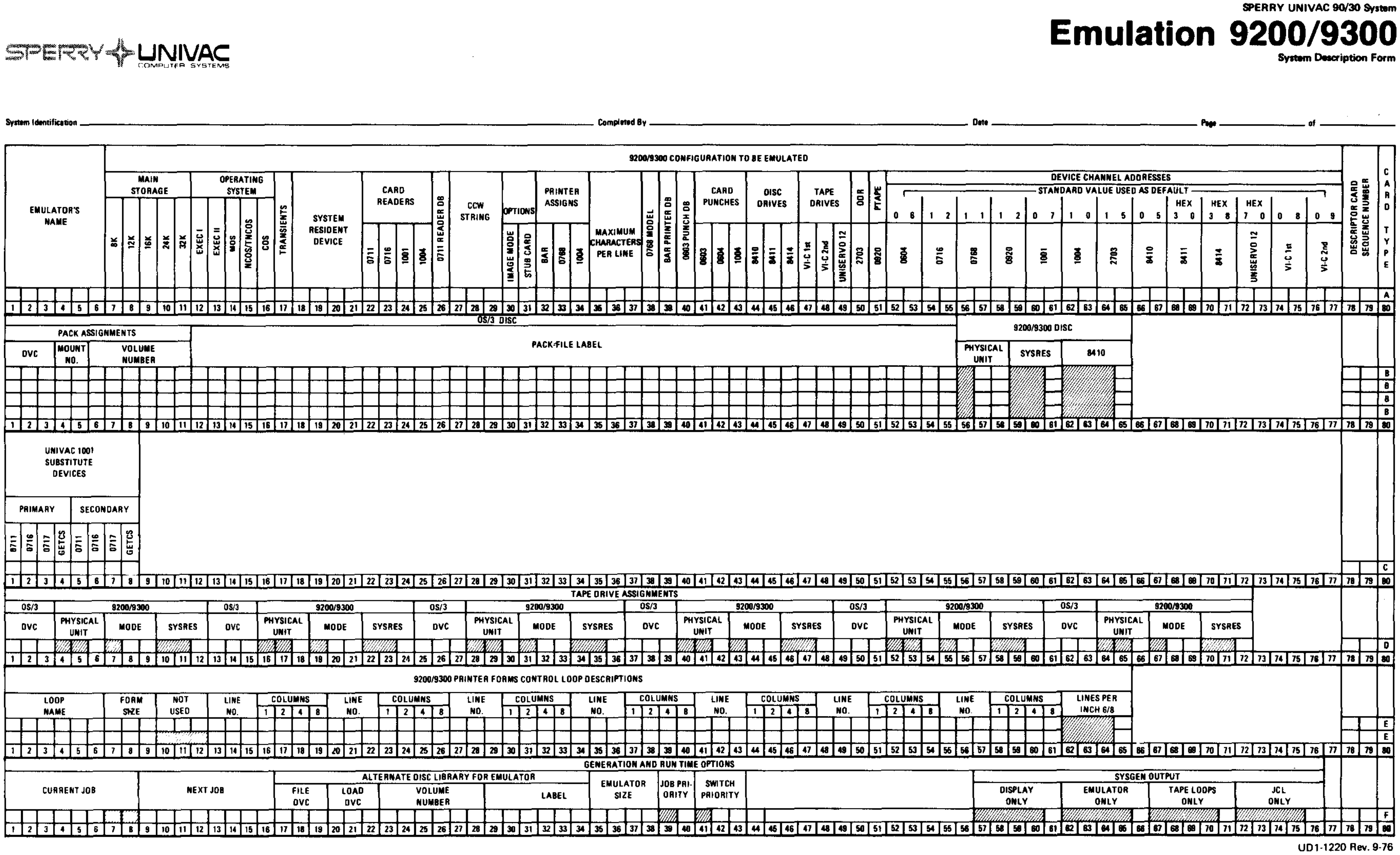

COMPLETE EMULATION SPECIFICATIONS ON UD-1220
PUNCH EMULATION SPECIFICATIONS INTO APPROPRIATE DESCRIPTOR CARDS
SUBMIT CARDS TO EMULAT PHASE
RU SG$PARAM DESCRIPTOR CARD VALIDATION
ERRORS?
YES
CORRECT CARD
NO
RU SG$EMUMK ASSEMBL/LINK EMULATOR AND/OR PRINTER LOOPS
RU ST$EMJCL GENERATE OS/3 JOB CONTROL DECK FOR EXECUTION OF EMULATED PROGRAM
EXECUTABLE EMULATOR LOAD MODULE
OS/3 JOB CONTROL DECK

Figure 5—1. Basic Steps In EMGEN Process



SPERRY✦UNIVAC
COMPUTER SYSTEMS

SPERRY UNIVAC 90/30 System

Emulation 9200/9300

System Description Form

System Identification ______ Completed By ______ Date ______ Page ______ of ______

EMULATOR'S NAME		1 2 3 4 5 6
9200/9300 CONFIGURATION TO BE EMULATED		
MAIN STORAGE	8K	7
	12K	8
	16K	9
	24K	10
	32K	11
OPERATING SYSTEM	EXEC I	12
	EXEC II	13
	MOS	14
	NCOS/TNCOS	15
	COS	16
TRANSIENTS		17
SYSTEM RESIDENT DEVICE		18 19 20 21
CARD READERS	0711	22
	0716	23
	1001	24
	1004	25
0711 READER DB		26
CCW STRING		27 28 29
OPTIONS	IMAGE MODE	30
	STUB CARD	31
PRINTER ASSIGNS	BAR	32
	0768	33
	1004	34
MAXIMUM CHARACTERS PER LINE		35 36 37
0768 MODEL		38
BAR PRINTER DB		39
0603 PUNCH DB		40
CARD PUNCHES	0603	41
	0604	42
	1004	43
DISC DRIVES	8410	44
	8411	45
	8414	46
TAPE DRIVES	VI-C 1st	47
	VI-C 2nd	48
	UNISERVO 12	49
ODR	2703	50
PTAPE	0920	51
DEVICE CHANNEL ADDRESSES — STANDARD VALUE USED AS DEFAULT	0 6 — 0604	52 53
	1 2 — 0716	54 55
	1 1 — 0768	56 57
	1 2 — 0920	58 59
	0 7 — 1001	60 61
	1 0 — 1004	62 63
	1 5 — 2703	64 65
	0 5 — 8410	66 67
	HEX 3 0 — 8411	68 69
	HEX 3 8 — 8414	70 71
	HEX 7 0 — UNISERVO 12	72 73
	0 8 — VI-C 1st	74 75
	0 9 — VI-C 2nd	76 77
DESCRIPTOR CARD SEQUENCE NUMBER		78 79
CARD TYPE	A	80

OS/3 DISC		
PACK ASSIGNMENTS	DVC	1 2 3
	MOUNT NO.	4 5
	VOLUME NUMBER	6 7 8 9 10 11
PACK-FILE LABEL		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
9200/9300 DISC	PHYSICAL UNIT	56 57 58
	SYSRES	59 60 61
	8410	62 63 64 65
		66 67 68 69 70 71 72 73 74 75 76 77
		78 79
	B B B B	80

UNIVAC 1001 SUBSTITUTE DEVICES		
PRIMARY	0711	1
	0716	2
	0717	3
	GETCS	4
SECONDARY	0711	5
	0716	6
	0717	7
	GETCS	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
	C	80

TAPE DRIVE ASSIGNMENTS																							
OS/3	9200/9300			OS/3	9200/9300			OS/3	9200/9300			OS/3	9200/9300			OS/3	9200/9300			OS/3	9200/9300		
DVC	PHYSICAL UNIT	MODE	SYSRES	DVC	PHYSICAL UNIT	MODE	SYSRES	DVC	PHYSICAL UNIT	MODE	SYSRES	DVC	PHYSICAL UNIT	MODE	SYSRES	DVC	PHYSICAL UNIT	MODE	SYSRES	DVC	PHYSICAL UNIT	MODE	SYSRES
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 | D 80

9200/9300 PRINTER FORMS CONTROL LOOP DESCRIPTIONS																	
LOOP NAME	FORM SIZE	NOT USED	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINE NO.	COLUMNS 1 2 4 8	LINES PER INCH 6/8
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 | E E 80

GENERATION AND RUN TIME OPTIONS													
		ALTERNATE DISC LIBRARY FOR EMULATOR							SYSGEN OUTPUT				
CURRENT JOB	NEXT JOB	FILE DVC	LOAD DVC	VOLUME NUMBER	LABEL	EMULATOR SIZE	JOB PRIORITY	SWITCH PRIORITY		DISPLAY ONLY	EMULATOR ONLY	TAPE LOOPS ONLY	JCL ONLY
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 | F 80

UD1-1220 Rev. 9-76

Figure 5—2. Emulation 9200/9300 System Description Form

The fields and functions of this form are directly related to the format of the descriptor cards described in Section 6. You can then punch your descriptor cards according to the specifications recorded on the emulator system description form. Once prepared, the descriptor cards are used as input for emulator generation. The exact procedure for initiating the emulator generation process depends on whether EMGEN is to be part of the SYSGEN process or if it is to be conducted as a separate function. In both cases, however, the process uses the canned job control streams provided in the EMULAT section of your SYSRES pack.

A full description of the instructions and routines comprising the EMULAT phase is provided in the system installation guide, UP-8074 (current version).

If EMGEN is to be part of the system generation procedure, the descriptor cards are placed in the reader and submitted as part of the SYSGEN card deck. When the EMULAT phase of the SYSGEN procedure takes place, the descriptor cards are processed and validated and an executable emulator load module is generated according to your specifications. If you have requested an OS/3 job control deck for executing your emulated program, one will be prepared for you.

If, on the other hand, you wish to generate an emulator as a separate function, you submit the descriptor cards in the following manner: first, you organize your card deck in the sequence shown in Figure 6—1. Place the cards in the reader and initiate the EMULAT phase by a keyin from the system console. The exact procedures for initiating the emulator generation process and for using the canned emulator generation job streams are provided on your SYSRES. Refer to Sections 5 and 9 of the OS/3 system installation guide, UP-8074 (current version).

6. Descriptor Card Preparation

6.1. GENERAL

Descriptor cards are your communications link (interface) to that phase of the system generation software used for emulator generation (EMGEN). Through the use of these cards, you inform and describe, for EMGEN, all of the requirements necessary to custom tailor an executable emulator load module to handle the programs to be emulated.

You must prepare and submit to EMGEN a deck of descriptor cards each time you wish to generate a new emulator. They provide two classes of information to the generation process: first, they describe your previous 9200/9300 hardware configuration and software system needed for your emulated program. Second, they describe how your programs use the hardware/software configuration described.

The information presented in this section is devoted to discussions of each descriptor card type (types A through F) and their related fields. Included are functional descriptions, how entries are made, what function each field performs, and what happens if a field is omitted or specified incorrectly. Error conditions and their resultant diagnostic messages are provided for each field entry. An incorrect or invalid field entry means that the field contains an entry which, from the user's standpoint, is incorrect. The entry, however, may conform to the syntactical requirements for that particular field. Columns 78 and 79 of all the descriptor cards may be used for establishing a card sequence order. These columns are ignored by EMGEN.

To assist you in preparing your descriptor cards, it is suggested that you first outline your emulation requirements on the related emulation 9200/9300 description form UD1-1220 shown in Figure 5—2. Table 6—1 summarizes the function of each descriptor card and lists whether the information on a particular card is used to structure your emulator or to produce the OS/3 job control needed to execute your emulated programs. Figure 6—1 illustrates the sequence in which the descriptor cards must appear in your job stream to EMGEN.

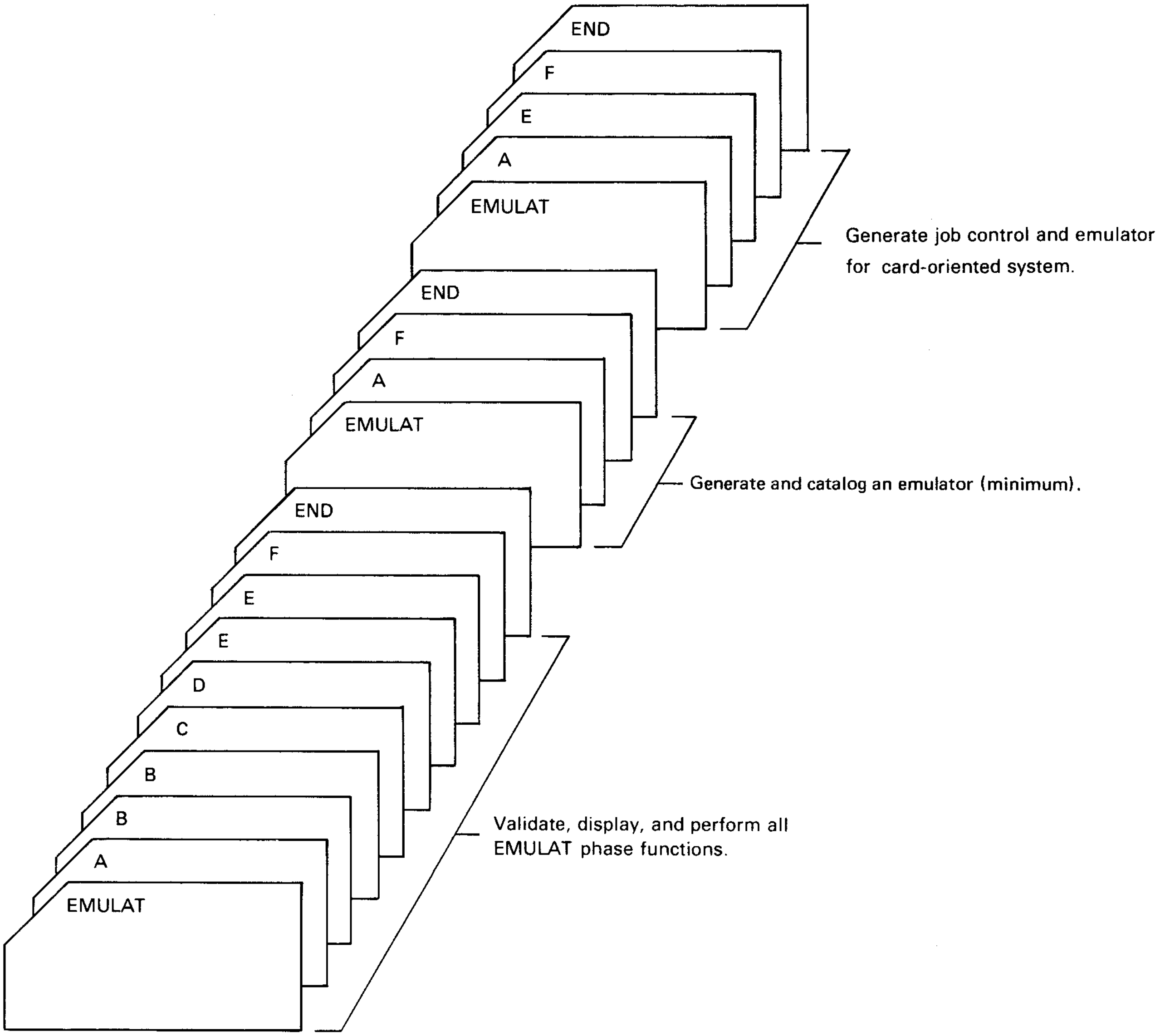
Table 6—1. 9200/9300 Descriptor Card Matrix

Card Type						Operation To Be Performed
A	B	C	D	E	F*	
E	J	E	J	J	J	Validate, display, and perform all SYSGEN functions.
E					J	Generate and catalog emulator.
E				J	J	Generate job control and emulator for card-oriented system.
E	J			J	J	Generate job control and emulator for disc-oriented system.
E			J	J	J	Generate job control and emulator for tape-oriented system.
E				J	J	Generate forms control data and job control for card-oriented system.

*See 6.7 for all available options

LEGEND:

E	Emulator oriented
J	Job control oriented


END
F
E
A
EMULAT
Generate job control and emulator for card-oriented system.
END
F
A
EMULAT
Generate and catalog an emulator (minimum).
END
F
E
E
D
C
B
B
A
EMULAT
Validate, display, and perform all EMULAT phase functions.

NOTES:

1. More than one descriptor card type B, D, or E may be submitted between each EMULAT and END card, depending on emulated programs requirements.

2. Each EMGEN descriptor grouping, between EMULAT and END, causes one execution of the EMULAT phase; output varies with descriptor cards submitted. It is necessary, however, to execute SG$PARAM for each descriptor card group.

3. The examples above are also referenced in Table 6—1.

4. The words EMULAT and END both start in column 1.

Figure 6—1. Sequence of Cards for Performing EMULAT Phase Functions

Another concern when preparing your descriptor cards is the limitation on the number of cards submitted for each execution of EMGEN and the sequence in which these cards appear in the job stream.

- Cards must be sequenced in ascending order by card type (specification in column 80).
- The type A card is mandatory unless a type F card with the TAPE LOOPS ONLY field (column 71) specified is included in the card deck to EMGEN.
- Type B and D cards must be sequenced in ascending order by DVC number. In addition, types B and D cards must have MOUNT numbers, sequenced in ascending order within DVCs.

  NOTE:

  The maximum number of disc or tape DVCs is 40.

- The maximum number of descriptor cards that may be submitted for each execution of EMGEN is:

Type A	1
Type C	1
Type B, D, and E	80 total (in any combination) of which no more than 25 may be type E cards with different loop names.
Type F	1

In the event that the total number of descriptor cards (type B, D, and E) exceeds the maximum allowable number (80), EMGEN terminates without further card validation and the following message is displayed.

NUMBER OF DESCRIPTOR CARDS EXCEEDS MAXIMUM PERMITTED

EMGEN also terminates and ceases validation if the descriptor card types are incorrectly specified (not A through F). In this case the following message is displayed.

DESCRIPTOR CARD NOT WITHIN PERMISSIBLE RANGE

6.2. DESCRIPTOR CARD TYPE A

The type A descriptor card describes the 9200/9300 hardware environment and software system required by the program or programs to be emulated. Only one type A card is permitted in the card sequence to EMGEN. The only time the type A card may be omitted from the card sequence to EMGEN is when the card sequence contains a type F descriptor card (6.7) on which the TAPE LOOPS ONLY field is specified.

Field Header: EMULATOR'S NAME
Card Column: 1—6
Card Type: A

Field Function:

Specifies the emulator's name. Also specifies the name used on the OS/3 //JOB control card if the CURRENT JOB field in the type F descriptor card is not specified.

An entry in this field is mandatory.

How Specified:

Alphanumeric entry (A—Z, 0—9), left-justified. Must begin with an alphabetic and must consist of at least two characters. Embedded blanks not permitted. Trailing blanks are zero-filled by EMGEN.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid:	ENTRY MUST BEGIN WITH AN ALPHA CHARACTER, BE LEFT JUSTIFIED, AND CONTAIN NO EMBEDDED BLANKS.
Field entry contains less than two characters:	NAME MUST CONTAIN AT LEAST TWO VALID CHARACTERS.

Field Header: MAIN STORAGE
Card Column: 7—11
Card Type: A

Field Function:

Indicates the main storage capacity of your SPERRY UNIVAC 9200/9300 System.

An entry in this field is mandatory.

How Specified:

With an X in the appropriate column of the field. Only one entry required. Multiple entries not permitted.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Multiple field entries specified	FIELD CONTAINS MORE THAN THE ONE ENTRY PERMITTED.

Field Header:	OPERATING SYSTEM
Card Column:	12—16
Card Type:	A

Field Function:

Specifies which operating system was used on the 9200/9300.

EXEC I	Card system
EXEC II	Card system
MOS	Card resident system
NCOS/TNCOS	Disc, tape, or transient NCOS system
COS	Disc or tape system

An entry in this field is mandatory.

How Specified:

With an X in the appropriate column of the field. Only one entry required. Multiple entries not permitted.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Multiple field entries specified	FIELD CONTAINS MORE THAN THE ONE ENTRY PERMITTED.

NOTE:

Entry in this field is checked against tape or disc specifications. Failure to assign either a tape or disc, the inclusion of tape or disc, or the absence of tape/disc descriptor cards could result in an error message due to incompatibility between operating system and the peripherals specified. (See 6.3 and 6.5.)

Card Column	*Operating System*	*Disc*	*Tape*
12	*Exec I*	*NO*	*NO*
13	*Exec II*	*NO*	*NO*
14	*MOS*	*OPTIONAL*	*OPTIONAL*
15	*NCOS, Transient NCOS*	*OPTIONAL*	*OPTIONAL*
16	*COS*	*OPTIONAL*	*OPTIONAL*

Field Header: TRANSIENTS
Card Column: 17
Card Type: A

Field Function:

Specifies that emulator is to be generated with critical functions as resident code. (The emulator normally is generated with critical functions in transients.) You can save approximately 1800 bytes of main storage by performing critical functions via OS/3 transients.

How Specified:

With an R. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

This field is not validated.

NOTE:

A smaller, slower version of the emulator is generated when this entry is not specified.

Field Header: SYSTEM RESIDENT DEVICE
Card Column: 18—21
Card Type: A

Field Function:

Indicates the 9200/9300 device used for loading the supervisor and job control.

An entry in this field is mandatory.

How Specified:

As:

0711
0716
1001
1004
8410
8411
8414
TAPE

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	MANDATORY ENTRY OMITTED.

Field Header: CARD READER
Card Column: 22—25
Card Type: A

Field Function:

Indicates the card reader or readers required by the program to be emulated.

An entry in this field is mandatory if a reader is to be supported by the emulator generated.

How Specified:

With an X in the appropriate column for each reader used on your 9200/9300 system. Multiple entries are permitted.

Possible Validation Errors and Resultant Diagnostic Messages:

0711 field entry omitted	MANDATORY ENTRY OMITTED.

NOTES:

1. *You may generate a single emulator that supports both readers and punches, and you may run under that emulator jobs that do not require either a reader or a punch. In order to free the 90/30 readers and punches for programs not requiring their use, you must generate a special JCL which causes both the reader and punch fields of this card to be ignored (not validated). You generate the JCL by specifying the JCL ONLY field on the type F descriptor card.*

2. *The COS, NCOS, and TNCOS Systems assume that 9200/9300 job and control streams will be read from the 0711 reader. If one of these systems is specified and the 0711 is not specified in column 22, the warning OPERATING SYSTEM USES 0711 AS CONTROL STREAM READER is issued. The omission is not treated as a fatal error because the emulation user can embed the 9200/9300 job and control stream in the OS/3 JCL which EMGEN creates to execute the emulator.*

   In order to accommodate 9200/9300 job and control streams as embedded data, the emulator must be told of its presence and OS/3 job control must be told to ignore slash-asterisk (/) found in the 9200/9300 deck.*

To use the embedded data approach, in lieu of the physical reader:

a. Prepare an OS/3 JCL card: // OPTION SCAN

b. Insert this card immediately after the // JOB card in the deck produced by EMGEN.

c. Remove and discard the special emulator delimiter, =/, from the OS/3 JCL deck and, in its place, insert the 9200/9300 deck. The special emulator control cards, identified by a B, D, G, or P in column 80, must be AHEAD of the 9200/9300 deck.*

d. Remove each / card from the 9200/9300 deck and replace it with a // CR card.*

e. Put all / cards, removed from the 9200/9300 deck in the card reader. Follow each /* card with a // CRFIN card.*

3. *You must include a type C descriptor card (6.4) if you have specified the 1001 card reader.*

Field Header: 0711 READER DB
Card Column: 26
Card Type: A

Field Function:

Causes emulator to provide two buffers for 0711 reader.

How Specified:

With an X. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Device to be double buffered not specified	MANDATORY ENTRY OMITTED.
IMAGE MODE field (column 30) and double buffering for 0711 reader specified	IMAGE MODE INCOMPATIBLE WITH DOUBLE BUFFERING

NOTES:

Because the emulator runs one card image ahead of the emulated program, user intervention is required in the event of a read error. If a card read error occurs, OS/3 displays an advisory message. You must remove and correct the card in error, then place it and all the cards that followed it into the reader's feed station. Once the cards are properly positioned, key in the directive:

```
(job#) COREAD R
```

The emulator will recover and proceed. If the R is omitted from the console entry, the emulator will display:

```
0254 INVALID COMMAND — FIRST OPERAND.
```

You must reenter a corrected command. (See COREAD command in Appendix B.)

Field Header:	CCW STRING
Card Column:	27—29
Card Type:	A

Field Function:

Specifies the maximum number of chained CCWs.

An entry in this field is mandatory if the emulator's default value (090) is unacceptable.

How Specified:

A numeric entry of 001 through 999. Leave blank if emulator's default value of 090 CCWs is acceptable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid (not numeric)	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.

Field Header:	OPTIONS
Card Column:	30—31
Card Type:	A

Field Function:

Indicates whether the emulated program uses image mode read or stub cards.

An entry in this field is mandatory if image mode or stub cards are used.

How Specified:

With an X in the appropriate column of the option wanted. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry specified but CARD READERS field entry omitted	IMAGE MODE AND STUB CARD REQUIRE READER SPECIFICATION.

NOTES:

1. *This field is ignored (not validated) if JCL-ONLY field is specified on the type F descriptor card.*

2. *Neither IMAGE MODE nor STUB CARD may be used if the 0711 reader or the 0603 punch is to be double buffered.*

Field Header: PRINTER ASSIGN'S
Card Column: 32—34
Card Type: A

Field Function:

Specifies the assignment of the emulated 9200/9300 printers to the 90/30 printers.

At least one entry in this field is mandatory.

How Specified:

By marking the appropriate column of the field with the specific character code required to relate the emulated 9200/9300 printer (being assigned) to the 90/30 printer desired. (See chart provided.)

Character Code	90/30 Printer
X	Any printer assigned
2	0773
4	0776
6	0768
8	0770
B	9300 BAR on channel*

*The 9200/9300 0768 printer may not be assigned to the BAR printer connected to the 90/30 via the channel adapter.

Possible Validation Errors and Resultant Diagnostic Messages:

No printer assigned	MANDATORY ENTRY OMITTED.
0768 printer assigned but entry in 0768 MODEL field (column 38) omitted	MANDATORY ENTRY OMITTED.
Field entry invalid	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.

— Entry does not conform to codes provided in chart

— 0773 printer assigned more than once

— BAR printer assigned more than once	
Printer assignment exceeds DVC capacity	CONFIGURATION DEFINED EXCEEDS STANDARD DVC LIST. CHECK OS/3 JCL PRODUCED BY EMGEN

NOTES:

1. *EMGEN produces the OS/3 JCL to execute emulated programs using the system standard DVCs and the LFDs required by the emulator. The system standard assigns two general DVCs and two specific DVCs for each printer except the channel-connected 9300 BAR printer. These standard DVCs are:*

20,21	*General (any printer)*
22,23	*0773*
24,25	*0776*
26,27	*0768*
28,29	*0770*

The emulator assumes that the 9200/9300 printers will always be assigned to the following LFDs:

PRNTR731	*BAR*
PRNTR681	*0768*
PRNTR141	*1004*

EMGEN, according to system requirements, generates specific DVCs before general DVCs. If, for example, column 32 contained an X, column 33 an 8, and column 34 a B, the JCL produced would be:

```
// DVC 28   // LFD PRNTR681    0768 on 0770
// DVC 21   // LFD PRNTR731    Bar on any printer
// DVC 20   // LFD PRNTR141    1004 on channel-connected BAR printer
```

Note that the BAR printer of the channel-connected 9300 utilizes one of the general DVCs.

With the exception of the 0773 (integrated) printer and the channel-connected BAR printer, it is possible to assign all three 9200/9300 printers to three 90/30 printers of the same type, thus exceeding the limits of standard DVCs. If this condition occurs, EMGEN will display the diagnostic message which tells you that you must review the JCL produced for compatibility with your requirements. (CONFIGURATION DEFINED EXCEEDS STANDARD DVC LIST. CHECK OS/3 JCL PRODUCED BY EMGEN.) It may be necessary to SYSGEN different DVCs for the 90/30 printer configuration. In any case, the OS/3 JCL deck must have the specific DVCs first and the LFDs, shown above, must be used to equate the 9200/9300 printer.

2. *EMGEN produces a // VFB card for each printer assignment. This card is required in all cases except when a 9200/9300 0768 printer is equated to a 90/30 0768 printer and the system is operated in a nonspooled mode. In this one instance, the emulator ignores use of the VFB and assumes that the user has prepared an appropriate carriage control loop for his 90/30 0768 printer. When the 0768-to-0768 printer assignment operates in a spooled environment, a VFB is required and that VFB must correspond to the 90/30 0768 control loop.*

   The VFB card, produced by EMGEN, specifies the emulator default, FORMNAME=STANLP00. This standard provides:

Form length	*66 lines*
Top line	*Line 1*
Bottom line	*Line 65*

   If this standard is unacceptable, you have two alternatives, both of which require generation of a special tape loop image. You may replace the VFB card or you may insert an L card in the JCL deck produced by EMGEN.

   - *Changing the VFB card*

     You may prepare your own // VFB card, by use of the FORMNAME= field to specify any name you desire. This card then replaces the one in the JCL deck. You could, of course, define a loop with the name STANLP which would, when filed by EMGEN, overlay the emulator standard. In this second approach, you need not replace the existing VFB card in the JCL deck.

   - *The L card*

     This is a special control card inserted in the emulator's embedded data stream to cause the emulator to bypass the standard VFB and substitute one designated by you on the L card.

     Format:

LABEL 1	△OPERATION△ 10	 16	OPERAND	△
L	loop-	name	9200/9300 printer type	

where:

L

Is the card type literal

loopname

Is the 8-character name of the loop to replace the standard VFB definition.

9200/9300 printer type

A BLANK for BAR printer; 1 for 0768; 2 for 1004.

The L card must be inserted after the / card and before the ?*/ card in the JCL deck generated. The JCL deck may or may not have B, G, or P cards between /* and ?*/. It does not matter where the L card occurs just so long as it appears between the delimiters noted.*

The LOOP (console) command may also be utilized to change the emulator's control of the 90/30 printer. It is not mentioned as an alternative, here, because the emulator must be stopped to accept a LOOP command. Loop-and-go programs may not permit stopping of the emulator at the point required.

3. *LOAD CODE commands, encountered in 9200/9300 software are bypassed in case the 0768 printer is being assigned a printer other than an 0768. If the user equates a 9200/9300 0768 printer to the 90/30 0770, 0773, or 0776 printers, and special characters must be defined, you must prepare an OS/3 // LCB statement and insert this statement into the JCL deck produced by EMGEN.*

4. *If the 0768 printer is assigned to any 90/30 printer, column 38 of the type A descriptor card must show the model of the 9200/9300 0768 printer.*

Field Header:	MAXIMUM CHARACTERS PER LINE
Card Column:	35—37
Card Type:	A

Field Function:

Indicates the maximum number of print positions required by the program to be emulated.

An entry in this field is mandatory unless the emulator supplied default value of 096 is acceptable.

How Specified:

A numeric entry of:

096
120
132

If more than one printer is specified, the number of print positions of the largest printer should be specified. Leave blank if default value of 096 characters is acceptable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.

Field Header: 0768 MODEL
Card Column: 38
Card Type: A

Field Function:

Indicates the model of the 0768 printer on your 9200/9300 system.

An entry in this field is mandatory if the 0768 printer is to be emulated.

How Specified:

With the appropriate character A, B, or C.

Possible Validation Error Codes and Resultant Diagnostic Message:

Field entry omitted and 0768 column (33) indicated in printer assigns field	MANDATORY ENTRY OMITTED.
Field entry specified but 0768 not specified in printer assigns field	MANDATORY ENTRY OMITTED.

Field Header:	BAR PRINTER DB
Card Column:	39
Card Type:	A

Field Function:

Causes emulator to provide two buffers for bar printer.

How Specified:

With an X. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Device to be double buffered not specified	MANDATORY ENTRY OMITTED.

Field Header: 0603 PUNCH DB
Card Column: 40
Card Type: A

Field Function:

Causes emulator to provide two buffers for 0603 punch.

How Specified:

With an X. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Device to be double buffered not specified	MANDATORY ENTRY OMITTED.
IMAGE MODE field (column 30) and double buffering for 0603 punch specified	IMAGE MODE IS INCOMPATIBLE WITH DOUBLE BUFFERING.

Field Header: CARD PUNCH
Card Column: 41—43
Card Type: A

Field Function:

Indicates whether the emulated program uses a card punch.

An entry in this field is mandatory if punches are to be supported by the emulator generated.

How Specified:

With an X in the appropriate column for each punch used in your 9200/9300 system. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

This field is not validated unless double buffering is specified for the 0603 punch.

Field Header: DISC DRIVES
Card Column: 44—46
Card Type: A

Field Function:

Indicates the type and quantity of disc drives required by the program or software to be emulated.

An entry in this field is mandatory if discs are used or if the operating system requires disc.

How Specified:

A numeric entry of 1—8 in the appropriate column of the field. Multiple entries are permitted. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry not numeric or out of range	ONLY NUMERICS 1 THRU 8 ARE PERMITTED.
Disc specified but no type B descriptor card provided	DISK DESCRIPTOR CARD OR RELATED FIELD ON THAT CARD OMITTED.
Entry not consistent with operating system requirements	TAPE/DISK INCLUSION/OMISSION IS INCOMPATIBLE WITH OPERATING SYSTEM SPECIFIED.
8410 specified on type B card but not on type A card	8410 SPECIFICATION ON B CARD REQUIRES CORRESPONDING SPECIFICATION ON CARD A.

Field Header:	Tape Drives
Card Column:	47—49
Card Type:	A

Field Function:

Indicates the type and quantity of tape drives required by the program or software to be emulated.

An entry in this field is mandatory if tapes are used or if the operating system requires tape.

How Specified:

A numeric entry of 1—8 in the appropriate column of the field. Multiple entries are permitted. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry not numeric or out of range limit	ONLY NUMERICS 1 THRU 8 ARE PERMITTED.
Disc specified but no type D descriptor card provided	DISK DESCRIPTOR CARD OR RELATED FIELD ON THAT CARD OMITTED.
Entry not consistent with operating system requirements	TAPE/DISK INCLUSION/OMISSION IS INCOMPATIBLE WITH OPERATING SYSTEM SPECIFIED.

Field Header: ODR/PTAPE
Card Column: 50/51
Card Type: A

Field Function:

Indicates whether the emulated program uses the 2703 optical documents reader (ODR) or 0920 paper tape reader (PTAPE).

An entry in the field is mandatory if the ODR or paper tape is to be supported by the emulator.

How Specified:

With an X in the appropriate field. Leave blank if not applicable.

Possible Validiation Errors and Resultant Diagnostic Messages:

This field is not validated; any entry is assumed correct.

Field Header:	DEVICE CHANNEL ADDRESSES
Card Column:	52—77
Card Type:	A

Field Function:

Creates an association, within the emulator, between the 9200/9300 channel assignments and those used on the 90/30 system.

An entry in this field is mandatory if the default value indicated in each column of the field is not acceptable.

How Specified:

A numeric entry, right-justified and zero-filled. The form UD1-1220 indicates whether the entry must be decimal or hexadecimal. Leave blank if default value listed is applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.
Associated device not specified	MANDATORY ENTRY OMITTED.

Field Header: CARD TYPE
Card Column: 80
Card Type: A

Field Function:

Identifies the descriptor card type.

An entry in this field is mandatory.

How Specified:

The mandatory entry for this field is the character A.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.
No type A descriptor cards submitted. (See note.)	FIRST DESCRIPTOR CARD MUST BE INCLUDED UNLESS LOOPS-ONLY IS SPECIFIED.

NOTE:

The type A descriptor card is mandatory unless a type F descriptor card (6.7) with the TAPE LOOPS ONLY field specified is included in the card sequence submitted to EMGEN.

6.3. DESCRIPTOR CARD TYPE B

This card provides EMGEN with the information needed to generate OS/3 job control cards that describe the disc used by the program to be emulated. If no discs are required for either data files or system residence, this card may be omitted.

One file may be described on each system description form. More than one type B card may be submitted; their total number, however, is limited according to the restrictions outlined in 6.1. If more than two cards are submitted, they must be entered in ascending sequence of DVCs. Cards within each group of DVCs must be arranged in ascending sequence by MOUNT number.

Field Header:	DVC
Card Column:	1—3
Card Type:	B

Field Function:

Denotes the logical unit number of the 90/30 disc drive on which the system or data files reside.

An entry in this field is mandatory.

How Specified:

A numeric entry, right-justified, zero-filled left, within the range of 000—256. Appendix D lists standard OS/3 DVCs.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.
Field entry exceeds 256	SPECIFICATION INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.
DVC out of sequence	SEQUENCE ERROR.

Field Header:	MOUNT NUMBER
Card Column:	4—5
Card Type:	B

Field Function:

Denotes the sequence in which you mounted your 9200/9300 disc packs on any given disc drive.

An entry in this field is mandatory.

How Specified:

A numeric entry, right-justified, within the range of 00—99. Numbering must begin with 00 and be assigned consecutively.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 PERMITTED.
MOUNT number out of sequence	SEQUENCE ERROR.

NOTE:

If the program requires two packs and your previous system included two disc drives, the mount numbers for both volumes would be 00. Higher numbers are used only if additional packs were mounted on the same disc drive during execution of the same job. A physical unit may only be related to one mount number.

Field Header: VOLUME NUMBER
Card Column: 6—11
Card Type: B

Field Function:

Specifies the volume serial number of the 90/30 disc pack on which the system or data files now reside.

An entry in this field is mandatory.

How Specified:

An alphanumeric entry (A—Z, 0—9), left-justified, with no embedded blanks.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	ENTRY MUST BE LEFT JUSTIFIED, ALPHANUMERIC, AND CONTAIN NO EMBEDDED BLANKS.

Field Header:	PACK-FILE LABEL
Card Column:	12—55
Card Type:	B

Field Function:

Indicates the OS/3 label of the pack file.

How Specified:

Any entry is acceptable. Emulator assumes entire field, starting with column 12, as the pack-file label. Therefore, your entry must start in column 12.

Possible Validation Errors and Resultant Diagnostic Messages:

This field is not checked.

Field Header: PHYSICAL UNIT
Card Column: 57—58
Card Type: B

Field Function:

Specifies the physcial unit number assigned to this volume in your 9200/9300 system.

An entry in this field is mandatory.

How Specified:

A numeric entry within the range of 00 through 07.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	ONLY NUMERICS 0 THRU 7 ARE PERMITTED.

NOTE:

Although the quantity of discs required is specified as 1 through 8 on the type A descriptor card, they are specified as 00 through 07 on the type B descriptor card in order to correspond to program's addressing.

Field Header: SYSRES
Card Column: 61
Card Type: B

Field Function:

Indicates that the disc pack, identified on this card, is the 9200/9300 system resident pack. Only one type B descriptor card may specify SYSRES.

An entry in this field is mandatory if the disc pack described is SYSRES.

How Specified:

With an X. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Conflict due to multiple SYSRES specification on type B or type D descriptor cards	SYSRES SPECIFICATION APPEARS MORE THAN ONCE.

NOTE:

Both the type B (disc) card and the type D (tape) card are checked for a SYSRES field entry. Duplicate entries for this field result in a validation error condition.

Field Header:	8410
Card Column:	65
Card Type:	B

Field Function:

Indicates that the pack-file identified on this card is an 8410.

How Specified:

With an X. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

8410 field not specified on type A card	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.

Field Header:	CARD TYPE
Card Column:	80
Card Type:	B

Field Function:

Identifies the descriptor card type.

How Specified:

The mandatory entry for this field is the character B.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.

6.4. DESCRIPTOR CARD TYPE C

This card identifies the 90/30 reader or GETCS (embedded data) function which replaces the reader in your 9200/9300 system. You must include a type C card if you have specified the 1001 reader on a type A descriptor card.

The C card may be used to specify the 0711 reader only if you have a channel-connected 9200/9300 system.

If more than one C card is submitted, the first card processed is treated as the only card submitted.

Field Header:	1001 SUBSTITUTE DEVICES
Card Column:	1—4/5—8
Card Type:	C

Field Function:

Specifies the 90/30 reader or OS/3 GETCS function (embedded in job stream) that is to replace the primary or secondary feed station for the 1001 reader.

An entry in this field is mandatory if the 1001 column is specified in the CARD READERS field of the type A card (6.2).

How Specified:

With an X in the appropriate field. One entry is required and both primary and secondary fields may be marked. Multiple entries are not permitted. Only one type C card should be submitted for this entry. If more than one C card is submitted, only the first card is accepted. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Entry invalid (duplicated)	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.
1001 not specified on type A card	1001 SPECIFICATION DEMANDS A C CARD. C CARD INCLUSION REQUIRES 1001 SPECIFICATION.

Field Header:	CARD TYPE
Card Column:	80
Card Type:	C

Field Function:

Identifies the descriptor card type.

An entry in this field is mandatory.

How Specified:

The mandatory entry for this field is the character C.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.

6.5. DESCRIPTOR CARD TYPE D

This card provides the EMGEN with the information needed to generate OS/3 job control cards that describe the tape drives used by the program to be emulated. If no tape drives are required for either data files or system residence, this card may be omitted.

Seven tape drives may be described on each card; only eight descriptions are permitted.

The entries in each tape description field are logically related. If one entry is correctly completed, all must be. Tape drive descriptions need not be entered contiguously.

The type D descriptor card entries must be arranged in ascending order by DVC.

Field Header:	OS/3 DVC
Card Column:	1—3, 13—15, 25—27, 37—39, 49—51, 61—63
Card Type:	D

Field Function:

Denotes the logical unit number of the 90/30 tape drive on which the system or data files reside.

An entry in this field is mandatory if its associated physical unit, mode, or SYSRES fields are specified.

How Specified:

A numeric entry, right-justified, zero-filled left, within the range of 000—256. Appendix D lists OS/3 DVCs.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.
Field entry exceeds 256	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.
DVC out of sequence	SEQUENCE ERROR.

Field Header:	9200/9300 PHYSICAL UNIT
Card Column:	6, 18, 30, 42, 54, 66
Card Type:	D

Field Function:

Specifies the physical unit number of the tape drive used in your 9200/9300 system.

An entry in this field is mandatory if DVC, MODE, or SYSRES fields are specified on this card.

How Specified:

A numeric entry within the range of 0—7.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	ONLY NUMERICS 0 THRU 7 ARE PERMITTED.
Number of units exceeds 8	TAPE SPECIFICATION EXCEEDS MAXIMUM PERMITTED.

Field Header:	9200/9300 MODE
Card Column:	8—9, 20—21, 32—33, 44—45, 56—57, 68—69
Card Type:	D

Field Function:

Describes tape characteristics and recording mode.

An entry in this field is mandatory unless the system default specification established during supervisor generation for the tape drive used is correct.

How Specified:

With the applicable value selected from the mode column of Table 6—2. Leave blank if default specification is acceptable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.

Table 6—2. Tape Mode Characteristics

Tape	cc	Bytes per Inch	Parity	Translate Feature	Convert Feature
UNISERVO 12/16 Magnetic Tape Volumes					
7-track	10	200	odd	off	on
	20	200	even	off	off
	28	200	even	on	off
	30	200	odd	off	off
	38	200	odd	on	off
	50	556	odd	off	on
	60	556	even	off	off
	68	556	even	on	off
	70	556	odd	off	off
	78	556	odd	on	off
	90	800	odd	off	on
	A0	800	even	off	off
	A8	800	even	on	off
	B0	800	odd	off	off
	B8	800	odd	on	off
9-track	C3	800	odd	off	off
	80*	1600	odd	off	off
UNISERVO VI-C Magnetic Tape Volumes					
7-track	10	200	odd		on
	20	200	even		off
	30	200	odd		off
	50	556	odd		on
	60	556	even		off
	70	556	odd		off
	90	800	odd		on
	A0	800	even		off
	B0	800	odd		off
9-track	80	800	odd		off

*Also applies to the UNISERVO 20 Magnetic Tape Subsystem

NOTE:

The mode always must be specified for tape devices with phase-encoded capability.

Field Header:	9200/9300 SYSRES
Card Column:	12, 24, 36, 48, 60, 72
Card Type:	D

Field Function:

Indicates the tape drive on which the system resident tape will be mounted. Only one SYSRES specification is permitted.

An entry in this field is mandatory if SYSRES is on tape.

How Specified:

With an X. Leave blank if not SYSRES.

Possible Validation Errors and Resultant Diagnostic Messages:

Conflict due to duplicate SYSRES specifications on type D and type B descriptor cards.	SYSRES SPECIFICATION APPEARS MORE THAN ONCE.

NOTE:

Both the type D (tape) card and the type B (disc) card are checked for a SYSRES field entry. Duplicate entries for this field result in a validation error condition.

Field Header:	CARD TYPE
Card Column:	80
Card Type:	D

Field Function:

Identifies the descriptor card type.

An entry in this field is mandatory.

How Specified:

The mandatory entry for this field is the character D.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.

6.6. DESCRIPTOR CARD TYPE E

This card describes the printer control loops that you used in your previous system. If you use the 0768 printer on your 90/30 system, this card may be omitted. If more than one tape loop requires definition, you may describe a maximum of 25 loops for each execution of EMGEN.

The number of E cards submitted are limited according to the restrictions outlined in 6.1. If a line of a given form is defined more than once, the last definition received overlays any previous definition. If the card is omitted, default values are substituted. These values are:

Form length:	66 lines
Top line:	Line 1
Bottom line:	Line 65

For additional information concerning the printers, refer to the notes listed under the PRINTER ASSIGN'S field on the type A card (6.2).

Field Header:	LOOP NAME
Card Column:	1—6
Card Type:	E

Field Function:

Assigns a name to a printer control loop.

This field is mandatory for each new loop definition.

How Specified:

An alphanumeric entry (A—Z, 0—9), left-justified, with no embedded blanks. The first character must be alphabetic.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	ENTRY MUST BEGIN WITH AN ALPHA CHARACTER, BE LEFT JUSTIFIED, AND CONTAIN NO EMBEDDED BLANKS.

NOTE:

This name is entered by the 90/30 operator when a program being emulated requires a change of form control. The first E card must contain a loop name. Definitions that apply to that name must be entered before a subsequent loop name occurs. If, for multiple E cards, the name remains the same, it need not be repeated on each card. If it is repeated, it is tested against the last name received and is discarded if equal.

Field Header: FORM SIZE
Card Column: 7—9
Card Type: E

Field Function:

Defines the form length in number of lines.

An entry in this field is mandatory for each new loop definition.

How Specified:

A numeric entry, right-justified, within the range of 001—132.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry above 132	NUMBER OF LINES SPECIFIED EXCEEDS MAXIMUM PERMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.

NOTE:

When building the tape loop, EMGEN sets a skip code (X'07') for the first and last line of the form. This is accomplished to maintain compatibility with the vertical format buffer requirements of OS/3. The first type E card submitted for each new tape loop must contain the form size.

Field Header:	LINE NO.
Card Column:	13—15, 20—22, 27—29, 34—36, 41—43, 48—50, 55—57
Card Type:	E

Field Function:

Indicates the number of the line associated with the channel punching described in the adjacent COLUMNS field of this card.

An entry in this field is mandatory if the adjacent COLUMNS field is specified on this card.

How Specified:

A numeric entry, right-justified, within the range of 001—132 (form length). Each line specification must be associated with a channel punch specification (COLUMNS field) and the combined fields must appear contiguously. The first line of the form is referred to as 001. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry above 132	NUMBER OF LINES SPECIFIED EXCEEDS MAXIMUM PERMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.

Field Header: COLUMNS
Card Column: 16—19, 23—26, 30—33, 37—40, 44—47, 51—54, 58—61
Card Type: E

Field Function:

Indicates the channel or channels punched for a given line.

An entry in this field is mandatory if the adjacent, lower-numbered LINE NO. field is specified on this card.

How Specified:

With an X for normal skip lines and with the literal OVER for the overflow line. One entry is required. Multiple entries are permitted. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.

NOTE:

Because a special code must be generated in order to conform with OS/3 vertical format buffer control, you must specify the overflow line with the literal OVER.

Field Header:	LINES PER INCH 6/8
Card Column:	65
Card Type:	E

Field Function:

Specifies the number of lines printed per inch.

An entry in this field is mandatory if the system default of 6 lines per inch is not applicable.

How Specified:

An entry of X, 6, or 8. Leave blank if default specification of 6 is acceptable.

Possible Validation Error Codes and Resultant Diagnostic Messages:

Field entry invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.

Field Header: CARD TYPE
Card Column: 80
Card Type: E

Field Function:

Identifies the descriptor card type.

An entry in this field is mandatory.

How Specified:

The mandatory entry for this field is the character E.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.

6.7. DESCRIPTOR CARD TYPE F

This card is a multifunction descriptor. The options offered on this card are:

- linking of the current job to the next job;
- alternate storage for the emulator;
- emulator size;
- job priority; and
- switch priority.

All entries are optional. EMGEN provides default values for JOB PRIORITY and SWITCH PRIORITY specifications if you elect not to provide the specifications for these fields.

Field Header: CURRENT JOB
Card Column: 1—6
Card Type: F

Field Function:

Provides the name that is used on the OS/3 job control card.

An entry in this field is optional.

How Specified:

An alphanumeric entry (A—Z, 0—9), left-justified, consisting of at least two characters with no embedded blanks. The first character must be alphabetic. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid	ENTRY MUST BEGIN WITH AN ALPHA CHARACTER, BE LEFT JUSTIFIED, AND CONTAIN NO EMBEDDED BLANKS.
Field entry less than two characters	NAME MUST CONTAIN AT LEAST TWO VALID CHARACTERS.

NOTE:

If this field is left blank, the emulator name specified on the type A descriptor card is used on the OS/3 // JOB card.

Field Header: NEXT JOB
Card Column: 9—16
Card Type: F

Field Function:

Provides the name of the next job to be executed from the OS/3 job library file ($Y$JCS). The next job may be another emulated program, a converted program, or an OS/3 element.

An entry in this field is optional.

How Specified:

An alphanumeric entry (A—Z, 0—9), left-justified with no embedded blanks. The first character must be alphabetic. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid	ENTRY MUST BEGIN WITH AN ALPHA CHARACTER, BE LEFT JUSTIFIED, AND CONTAIN NO EMBEDDED BLANKS.

Field Header: FILE DVC
Card Column: 17—19
Card Type: F

Field Function:

Tells EMGEN where to file the loadable version of the emulator.

An entry in this field is mandatory if the LOAD DVC, VOLUME, or LABEL field is specified on this card.

How Specified:

A numeric entry, right-justified, zero-filled left, within the range of 000—256. Appendix D lists OS/3 DVCs. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.
Field entry nonnumeric	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.
Field entry exceeds 256	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.

NOTE:

The fields that describe an alternate load library for the emulator, columns 17—34, are logically related. If one field is specified, all fields must be completed. If none are completed, it is assumed that the emulator will reside in $Y$LOD of the SYSRES volume.

If an alternate load library is specified for the emulator, you must regenerate emulators during subsequent OS/3 releases.

Field Header:	LOAD DVC
Card Column:	20—22
Card Type:	F

Field Function:

Provides the information required to produce OS/3 job control statements that will load the emulator from an alternate disc pack.

An entry in this field is mandatory if the FILE DVC field is specified on this card.

How Specified:

A numeric entry, right-justified, zero-filled left, within the range of 000—256. Appendix D lists OS/3 DVCs. Leave blank if not applicable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	MANDATORY ENTRY OMITTED.
Field entry is nonnumeric.	ONLY NUMERICS 0 THRU 9 ARE PERMITTED.
Field entry exceeds 256.	SPECIFICATION IS INCOMPATIBLE WITH OTHER FIELDS OR SYSTEM REQUIREMENTS.

NOTE:

The fields that describe an alternate load library for the emulator, columns 17—34, are logically related. If one field is specified, all fields must be completed. If none are completed, it is assumed that the emulator will reside in $Y$LOD of the SYSRES volume.

If an alternate load library is specified for the emulator, you must regenerate emulators during subsequent OS/3 releases.

Field Header:	VOLUME NUMBER/LABEL
Card Column:	23—28/29—34
Card Type:	F

Field Function:

Defines the volume serial number of the 90/30 disc pack which contains the emulator's alternate load library and the name for that library.

How Specified:

Any entry is valid since these fields are not validated.

An entry in this field is mandatory.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted	MANDATORY ENTRY OMITTED.

NOTE:

An entry acceptable to OS/3 job control is required for both fields if either the FILE DVC or LOAD DVC fields have been specified. The actual characters in the entry are not checked but an error is noted if either field is left blank.

The fields that describe an alternate load library for the emulator, columns 17—34, are logically related. If one field is specified, all fields must be completed. If none are completed, it is assumed that the emulator will reside in $Y$LOD of the SYSRES volume.

If an alternate load library is specified for the emulator, you must regenerate emulators during subsequent OS/3 releases.

Field Header: EMULATOR SIZE
Card Column: 35—38
Card Type: F

Field Function:

Sets the emulator size for both minimum and maximum.

An entry in this field is mandatory if the FILED DVC field is specified.

How Specified:

A hexadecimal entry that specifies the main storage needed by the emulated program. No entry is required unless an alternate load library is specified.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry omitted or invalid	MEMORY REQUIREMENT MUST BE SPECIFIED WHEN LOADING FROM AN ALTERNATE LIBRARY.

NOTE:

Emulator size may be specified whether or not an alternate library is used. The entry must be correctly specified if an alternate load library is indicated; otherwise, a diagnostic warning message is generated.

Field Header: JOB PRIORITY
Card Column: 40
Card Type: F

Field Function:

Defines the priority of the job as normal (N), high (H), or preemptive (P).

An entry in this field is mandatory if either high or preemptive priority is required.

How Specified:

As N, H, P, or left blank, in which case EMGEN defaults to a priority of N.

Possible Validation Errors and Resultant Diagnostic Message:

Field entry invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.

Field Header:	SWITCH PRIORITY
Card Column:	42—43
Card Type:	F

Field Function:

Defines the order in which control is passed from task to task within the job or from job to job.

An entry in this field is mandatory if a priority other than 04 is required.

How Specified:

An alphanumeric value, right-justified, within the range of 01—62. Leave blank if the default priority is acceptable.

Possible Validation Errors and Resultant Diagnostic Messages:

Field entry invalid	SPECIFICATION IS INVALID — DEFAULT VALUE SUBSTITUTED.

Field Header: DISPLAY ONLY/EMULATOR ONLY/TAPE LOOPS ONLY/JCL ONLY
Card Column: 61, 66, 71, 76
Card Type: F

Field Function:

Defines the selected output to be produced by EMGEN.

How Specified:

With an X in the appropriate field for the output to be produced. Multiple entries are permitted. EMGEN will not produce a particular output, provided that the field for that output selection is left blank and at least one other output selection is specified. If all output fields are left blank, EMGEN produces an output for each output category. EMGEN validates the descriptor cards in all cases, except as noted.

NOTES:

1. *The type 1 descriptor card is not validated whenever TAPE LOOPS ONLY is specified.*

2. *The CARD READ and PUNCH fields on the type 1 descriptor card are ignored whenever the JCL ONLY field is specified.*

Field Header:	CARD TYPE
Card Column:	80
Card Type:	F

Field Function:

Identifies the descriptor card type.

An entry in this field is mandatory.

How Specified:

The mandatory entry for this field is the character F.

Possible Validation Errors and Resultant Diagnostic Messages:

Card out of sequence	SEQUENCE ERROR.

NOTE:

The type F descriptor card is required with the TAPE LOOP ONLY field specified whenever the type A descriptor card (6.3) is omitted from the card sequence submitted to EMGEN.

7. Operating Considerations

7.1. GENERAL

Successful execution of your programs requires that you have not only prepared your programs properly, but that you also have taken into consideration the various operating requirements, restrictions, and procedures necessary for using the emulation aids presented in this manual. This section discusses many of the operating considerations you should be aware of prior to executing your programs under emulation.

7.2. DEFINING EMULATOR MAIN STORAGE REQUIREMENTS

The emulator system description form provides four card columns labeled EMULATOR SIZE on the type F descriptor card for specifying the emulator's main storage requirements. This field is needed because OS/3 requires main storage stipulation for any program that is not loaded from either the system resident pack (SYSRES) or from the temporary job run library ($Y$RUN). EMGEN uses this field for both the minimum and maximum specifications on the // JOB card.

The EMULATION SIZE field may be used whether or not you elect to load from an alternate library. If an alternate library is specified, an entry should be included in order to punch a usable // JOB card. If an alternate library is specified and the main storage size is omitted, EMGEN will issue a warning message but will not count an error. This approach is taken because you cannot determine main storage requirements until after the link of the assembled emulator.

If you do not enter a main storage requirement and do specify an alternate library, you must later modify the // JOB card produced by EMGEN.

7.3. RUNNING SG$EMJCL UNDER SPOOLING

If SG$EMJCL is run in the spooling mode, the emulator JCL cards produced will be preceded by two blank cards and a job accounting card. These cards must be removed before using the deck to execute the emulator.

7.4. 9200/9300 VSN DUPLICATION FOR 8411/8414 EMULATION

To be emulated, the 8411 and 8414 discs must be physically connected to the 90/30. No provision is made for transferring pack file images from these packs to other 90/30 packs. In addition, the 9200/9300 software does not check the 8411 and 8414 VSNs and will, if not otherwise directed, prep these packs with a default VSN of '123456'.

OS/3, on the other hand, checks the VSNs and will not accept duplicates. If you alter the VSNs of these packs, leave those packs with duplicate VSNs offline until OS/3 directs their mounting. OS/3, at this time, will be past the point where it checks for duplicate VSNs. Remember, discs may not be prepped under emulation.

7.5. RESTRICTIONS FOR DISC PREP

Emulation 9200/9300 does not support the 9200/9300 disc prep routine due to the fact that the disc prep is not written to operate under the standard 9200/9300 software. Discs may be prepped on the 9200/9300 and used under emulation, or prepped under OS/3 and used under emulation. They may not, however, be prepped under emulation.

7.6. RELEASING PERIPHERALS NOT REQUIRED FOR PROGRAM BEING EMULATED

If the program being emulated does not require the punch reader peripherals, you may free them for assignment to other OS/3 programs by removing their respective JCL cards from the run deck produced during emulator generation.

You also have the option of reexecuting EMGEN to omit punching these JCL cards by omitting the appropriate specification for the device type on the type A descriptor cards, and by specifying JCL ONLY on the type F descriptor card. This will produce a second JCL deck similar to the original deck with the JCL cards omitted for devices not required.

7.7. SYSTEM HANG BECAUSE OF EMULATOR

When the emulator issues a message to the console, it will loop until the user supplies a response. This condition could cause the system to hang if the emulator is running at a higher priority than other jobs. To obviate this possibility, either answer messages promptly or lower emulator priority on the // EXEC card.

7.8. EMULATOR HANG BECAUSE OF / FINIS CARD

A hang condition will occur with the use of the / FINIS card in the 9200/9300 control stream. As on the 9200/9300 system, the program directive OP REQ E0 must be executed from the system console to allow further reading from the 9200/9300 control stream. If the emulator is to be terminated, the card immediately following the / FINIS card in the control stream must be a / PAUSE card (/ PAUSE 6FFF). If another 90/30 job or job step is to be executed next, the / PAUSE F666 card should be followed by another / PAUSE F666 card. When message HALT 6FFF is displayed on the console, a keyin of 10 EOJ will cause an orderly closing of any 90/30 file and will terminate the emulator. In summary, it is recommended that a pause card be used for emulator termination rather than the / FINIS card so as to avoid an emulator hang condition.

PART 4. APPENDIXES

Appendix A. Emulation Aids

A.1. THE 8410 EMULATION AID

The 8410 emulation aid is a software element that is used to transcribe the entire contents of your 8410 disc packs, a medium which cannot be read on the SPERRY UNIVAC 90/30 System, to any 90/30 disc pack supported by system access technique (SAT). (The 90/30 system supports the 8411, 8414, 8416, and 8430 disc units.) In this manner, your existing 9200/9300 system and data files will reside on a storage medium that can be accommodated by the 90/30 system which, through emulation, can execute the instructions, conventions, and interfaces of your programs under control of the SPERRY UNIVAC Operating System/3 (OS/3).

The primary goal of the 8410 emulation aid is to reduce, as much as possible, the programming complexities usually involved in transcribing data files from storage devices supported by one processing system but not supported by the other processing system. To accomplish this, the 8410 emulation aid uses two software programs: your existing 9200/9300 DUMP/RESTORE utility routine and the PIMAGE routine. (See Figure A—1.)

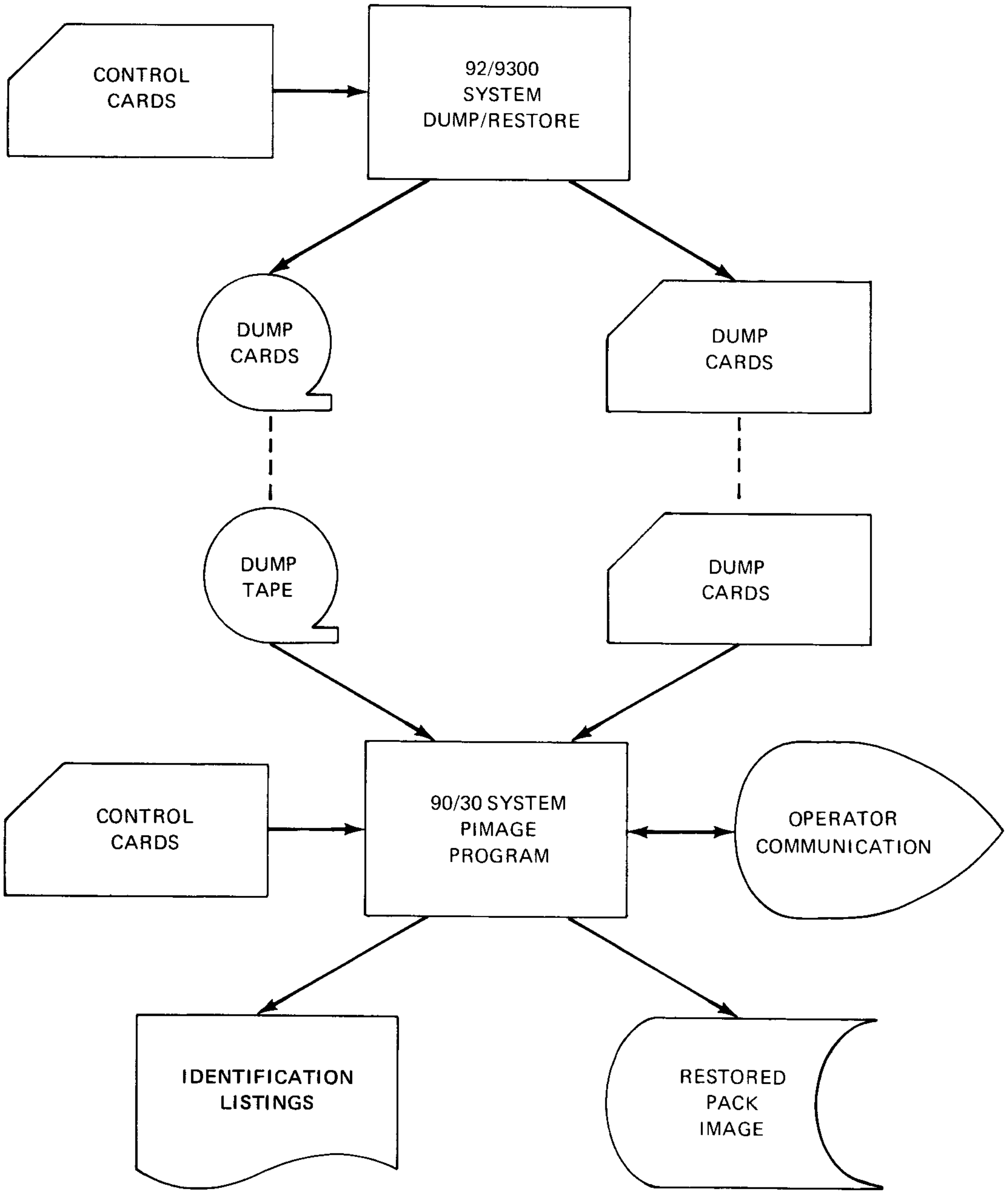


Figure A—1. Overview of 8410 Disc Transfer to 90/30 Disc

The DUMP/RESTORE routine allows you to dump your existing system to an intermediate storage device, namely, magnetic tape. This is the first step in transcribing your files to a 90/30-supported disc. The dump to tape or cards is necessary. Once you have completed a dump of your system, you may then restore a bit-for-bit image of your files to a 90/30-supported disc. The restore process is performed by the OS/3 PIMAGE routine. PIMAGE is a JPROC that, when called from the OS/3 system console, selects the emulator as well as the 9200/9300 RESTORE routine to be emulated. Your system files, which reside on an intermediate storage medium, are used as the 9200/9300 input to the same DUMP/RESTORE routine now running under emulation on the 90/30 system. Use of PIMAGE simplifies the image transfer process by eliminating the need for you to prepare an emulator. Your responsibilities can be summarized as:

1. Preparing the control cards necessary to initiate and dump your 9200/9300 system files to tape.

2. Determining the placement of your system files on the 90/30 disc, if the contents of more than one 8410 disc are to be contained on a physical 90/30 disc.

3. Preparing the control cards needed to run the PIMAGE program and initiating the execution of PIMAGE.

4. Making certain that each phase of PIMAGE has executed properly and deciding either to continue or end the job at the completion of phase 1.

5. Checking the output produced to make certain that the job was successfully executed.

It should be noted that the DUMP/RESTORE routine, when performing a dump of your system files, is executed on the 9200/9300 as a native mode program. When executed under OS/3, however, this same routine appears as data to the 9200/9300 emulator while the emulator appears as a problem program to OS/3. No special interfaces are required from OS/3 and the data base involved is that of the 9200/9300 system and your data files.

The PIMAGE routine requires a minimum main storage area of 49K bytes and may be executed on a minimal system equipped with a card reader and printer. If tape is used as the intermediate storage medium to which the dump was written, then the 90/30 system must also be equipped with at least one tape drive of the same type supported by the emulator used at the time the DUMP/RESTORE routine was executed.

Two types of card input are required to execute the PIMAGE routine: the PIMAGE control cards and the 9200/9300 control cards required as input to the RESTORE routine of the DUMP/RESTORE program. These cards are described in A.1.2.1, and in the 8410 disc subsystem utility programs, UP-7668 (current version), respectively. Because the 9200/9300 software dumps the 8410 on a file basis, it is impossible to run the emulated DUMP/RESTORE program without files being specified. To create an empty pack image (one that is equivalent to a prepped 8410 pack), only the PIMAGE control cards are used. You may then run the usual 9200/9300 programs such as VTOC to construct the contents of the pack image.

A.1.1. Dumping Your 8410 Discs to Tape or Cards

The control cards needed to dump your system files to an intermediate storage device are:

- Header card
- Dump limit card
- End card .

The control card input to the DUMP/RESTORE routine may be from either the card reader or the primary feed of the card controller when executed in a minimum operating system. In the operating system, however, the card input must be from the job control stream. In either case, the card output is to the serial or row punch. All such options are selected by incorporating the appropriate IOCS routines at the time you link the DUMP/RESTORE routine. It is also required that the DUMP/RESTORE routine be used with an operating system containing a disc dispatcher and, when tape is used as the intermediate storage medium, with a tape dispatcher. The DUMP/RESTORE routine will provide you with a printout indicating the addresses of those disc sectors that could not be read during the dump. It will also display on the printer messages concerning any errors that occurred during execution of the dump.

An illustration of the control cards needed for a dump operation and the sequence in which these cards should appear is shown in Figure A—2.

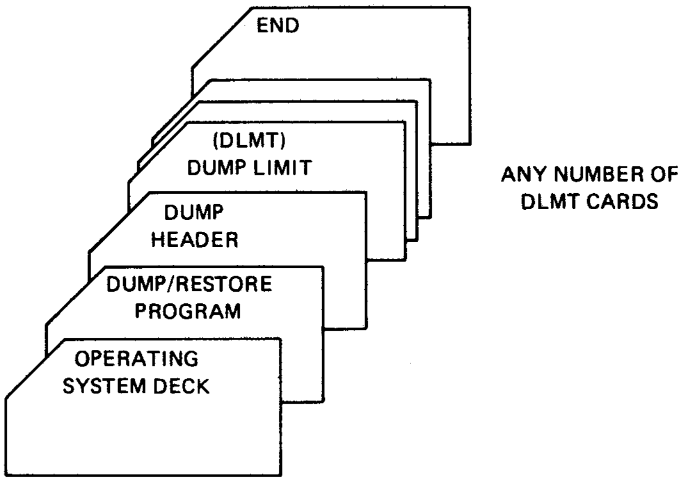


Figure A—2. Dump Control Card Sequence

A.1.1.1. Header Card

The header card defines the type of operation to be performed and specifies the intermediate storage device to which the contents of your 8410 disc are written. Only one header card is needed for the dump operation. The format of the header card is described in Table A—1.

Table A—1. Header Card Format

Card Columns	Contents	Meaning
10—13	DUMP	Indicates a dump operation is to be performed. The dump card is followed by one or more DMLT cards.
15	C or T	Indicates intermediate storage device is: C — Punched cards T — Magnetic cards
17—19	nnn	nnn is a 3-digit decimal number indicating the total number of control cards including the header and the end cards. This field is ignored on header cards after the first one.

A.1.1.2. Dump Limit Card

The dump limit card (DLMT) identifies the area on your 8410 disc to be transferred to the intermediate storage medium. Any number of DLMT cards may follow the header card. At least three DLMT cards must be prepared if you are dumping the entire 8410 disc. The three cards specify disc, Fastband, and volume table of contents (VTOC). The format of the DLMT card is described in Table A—2.

Table A—2. Dump Limit Card Format (Part 1 of 2)

Card Columns	Contents	Meaning
10—13	DLMT	Identifies the card type
15—16	nn	nn is the logical unit number of the input disc.
19—26	File name (eight alphanumeric characters)	The name of the file containing sectors to be dumped
	FASTBAND	FASTBAND to be dumped
	VTOCUPDT	VTOC to be dumped
29—32	ttss	The ttss is the address of the track and sector where the dump operation is to begin: tt is a 2-digit track address. ss is a 2-digit sector address.

Table A—2. Dump Limit Card Format (Part 2 of 2)

Card Columns	Contents	Meaning
	FBnn	Indicates a dump of the Fastband beginning with the sector specified by nn, a 2-digit number
	VTOC	Indicates that all or part of the VTOC is to be dumped
35—38	ttss	The ttss is the address of the last sector to be dumped: tt is a 2-digit track address. ss is a 2-digit sector address.
	FBnn	Indicates the last sector of the Fastband to be dumped: nn is a 2-digit sector address.
	VTnn	Indicates the last sector of the VTOC to be dumped: nn is a 2-digit sector address.
	VTOC	Indicates that all of VTOC is to be dumped
41—44	nn	nn is a 2-digit value indicating the logical unit number of the intermediate storage device defined on the last header card.
45—52	File name (eight alphanumeric characters)	The file name associated with the intermediate storage device

A.1.1.3. END Card

The END card is the last card in the sequence and specifies the end of the dump control cards. The format of the END card is described in Table A—3.

Table A—3. End Card Format

Card Columns	Contents	Explanation
1—2	EE	Identifies an end card
10—12	END	Signals the last of the dump control cards

A.1.2. Restoring Your 8410 Disc Image to a 90/30 Disc

The restore process, as previously stated, entails the writing of your 9200/9300 files onto OS/3-supported discs. This process may be performed anytime after you have completed the dump of your 8410 discs. The data being restored usually constitutes one side of an 8410 disc. However, because of the large capacity of the 90/30 system discs, an entire 90/30 pack is not required to hold the contents of the 8410 disc. Therefore, the term *pack-file* is being introduced. It represents all the files of data, transferred from one side of the 8410 disc, that have become one file on a 90/30-supported disc. The transfer of data from the intermediate storage device to the 90/30 disc begins when you initiate the execution of the PIMAGE routine. You must, however, first prepare the control cards that define information, such as the disc type to which the data is written, the volume serial number, etc., to the PIMAGE routine. These cards are inserted into the reader before starting the PIMAGE program.

A.1.2.1. PIMAGE Control Card Preparation

The conventions for preparation of the PIMAGE control card are:

- The parameters may extend over more than one card.

- The continuation of the parameters onto the second and subsequent cards is indicated by placing an X in column 72 and beginning the second or subsequent card with a // (represents a blank space). The parameters may begin in any column after the //.

- Do not repeat the word PIMAGE on the second or subsequent cards.

- The parameters used on the PIMAGE control card are keyword parameters. A keyword parameter consists of a word or a code immediately followed by an equal sign, which is, in turn, followed by a specification. Keyword parameters can be written in any order in the operand field. Commas are required only to separate parameters.

- Capital letters, commas, equal signs, and parentheses must be coded exactly as shown.

- Lowercase letters and words are generic terms representing information that must be supplied by the user.

- Information contained within braces represents mandatory entries of which one must be chosen.

- Information contained within brackets represents optional entries that (depending upon program requirements) are included or omitted. Braces within brackets signify that one of the specified entries must be chosen if that parameter is to be included.

- An optional parameter that has a list of optional entries may have a default specification that is supplied by PIMAGE when the parameter is not specified by the user. Although the default may be specified by the user with no adverse effect, it is considered inefficient to do so. For easy reference, when a default specification occurs in the format delineation, it is printed on a shaded background. If, by parameter omission, PIMAGE performs some complex processing other than parameter insertion, it is explained in the parameter description.

Format:

LABEL	△OPERATION△	OPERAND
//	PIMAGE	VOL9030=vvvvvv $\left[\text{DISCDRV=}\begin{Bmatrix}\text{ddd}\\ \text{050}\end{Bmatrix}\right]$ $\left[\text{DSKTYPE=}\begin{Bmatrix}\text{8411}\\ \text{8414}\\ \text{8416}\\ \text{8430}\end{Bmatrix}\right]$ $\left[\begin{Bmatrix}\text{INPUT=CARD}\\ \text{TAPEDRV=}\begin{Bmatrix}\text{ttt}\\ \text{090}\end{Bmatrix}\end{Bmatrix}\right]$ [CNTRL=NOINPT] $\left[\text{PACKNAME=}\begin{Bmatrix}\text{aaaaaaaa}\\ \text{DISC8410}\end{Bmatrix}\right]$ $\left[\text{RERUN=}\begin{Bmatrix}\text{NO}\\ \text{YES}\end{Bmatrix}\right]$

VOL9030 Keyword Parameter:

VOL9030=vvvvvv
Specifies the 6-digit volume serial number of the 90/30 disc that is to contain the 8410 data. This keyword parameter has no default value.

DISCDRV Keyword Parameter:

DISCDRV= $\left\{\begin{matrix}\mathbf{ddd}\\ \mathbf{050}\end{matrix}\right\}$

Specifies the disc drive to be used for the output of the PIMAGE run. The 050 represents a general class of disc drives and allows the system to determine which specific drive is available for use; this information is indicated to the operator through a mount message. If the general class of disc is other than 050, you may define it by the three digits that represent the general class.

DSKTYPE Keyword Parameter:

DSKTYPE= $\left\{\begin{matrix}\mathbf{8411}\\ \mathbf{8414}\\ \mathbf{8416}\\ \mathbf{8418}\\ \mathbf{8430}\end{matrix}\right\}$

Specifies the type of 90/30 disc available on the system and may be 8411, 8414, 8416, 8418, or 8430. The default value is 8416. If the user system is configured with 8416 disc drives, this parameter need not be submitted.

INPUT Keyword Parameter:

INPUT=CARD

Specifies that the input data will be read from card reader. If this parameter is specified, all keyword parameters related to the tape devices are ignored (i.e., TAPEDRV).

TAPEDRV Keyword Parameter:

TAPEDRV= $\left\{\begin{matrix}\mathbf{ttt}\\ \mathbf{090}\end{matrix}\right\}$

Specifies the tape drive to be used for the DUMP tape. The value 090 represents a general class of tape drives and allows the system to determine which specific drive to use; the specific drive would then be indicated to the operator through a mount message. If the general class number for tape drives on the system is other than 090, the user may submit his value by specifying a 3-digit number on the control card.

CNTRL Keyword Parameter:

CNTRL=NOINPT

Specifies that PIMAGE is to create a pack file that is equivalent to a prepped 8410 disc (one without VTOC). ←

PACKNAME Keyword Parameter:

PACKNAME= { **aaaaaaaa** / **DISC8410** }

Is an OS/3 JCL operand (LBL statement) and indicates the label of a file. The label value will be stored in the FORMAT1 label area of the OS/3 VTOC. The parameter can be up to eight characters in length for a disc file.

RERUN Keyword Parameter:

RERUN= { **NO** / **YES** }

Indicates whether a previous job had allocated an area for a file with the same name as will be created by this job. This parameter is normally not submitted but, in cases of an unsuccessful run whereby the pack-file must be re-created, specify this parameter as YES.

A SCRATCH function will automatically be executed before the run begins.

If an area of disc has been allocated to a file, the area cannot be released for reuse until a SCRATCH function is performed on that file.

Example 1:

```
    LABEL      ΔOPERATIONΔ            OPERAND                          Δ              COMMENT
1               10      16
// PIMAGE E VOL9030=PAYR01,DSKTYPE=8418,PACKNAME=DSP011
```

Run the PIMAGE program using 8418 disc drives to restore an 8410 pack from tape. The name of the pack-file is DSP011. This data will now reside on a 90/30 disc pack labeled PAYR01. The operating system will indicate to the operator where to mount the tape and disc packs, since the general class default drive numbers were not changed.

Example 2:

```
// PIMAGE PACKNAME=INVMON,VOL9030=INV001
```

Restore an 8410 pack from tape to an 8416 disc. The 90/30 disc pack has a volume serial number of INV001; the pack-file name is INVMON.

The data from the tape will become a file on the 90/30 disc and have a label of INVMON. This file will be on the 90/30 disc pack with the volume serial number of INV001. The file area will be large enough to contain entire 8410 data. The system decides on the placement of the pack-file on the disc.

A.1.2.2. Initializing the PIMAGE Routine

To initialize the PIMAGE, you must mount your dump tape on a SPERRY UNIVAC tape subsystem (input to the PIMAGE routine), insert your control cards in the reader, and issue the command RUN△PIMAGE from the system console. This calls the PIMAGE routine and initializes execution of the tape verification (phase 1) of the routine. It should be noted that the PIMAGE control cards must be followed by a // FIN card.

A.1.3. Operational Phases of the PIMAGE Routine

A.1.3.1. Tape Verification — Phase 1

This phase of the PIMAGE routine allows you to verify whether the input tape (prepared during the dump of your 8410 disc) is the correct tape to be used as input for the subsequent phases of the PIMAGE routine. It does this by printing out the file identification number and the file creation date so that you can determine if the proper tape is mounted. It also requests a reply of YES or NO from the system console to determine whether to continue with the process of restoring your 8410 image to a 90/30 disc or to terminate the job. In this manner, the PIMAGE routine also may be used to identify tapes in cases where there is a labeling problem.

A.1.3.2. Disc Initialization — Phase 2

This phase preps the appropriate area of the 90/30 disc to which the contents of the input dump tape are to be written. The exact amount of space needed varies with the specific 90/30 disc type being used. The entire allocated area is formatted into blocks of 160 bytes at allocation time. This preformatting is necessary so that the area can be accessed by the emulation program in conjunction with SAT. This step is skipped for the 8416 discs.

A.1.3.3. Data Restore — Phase 3

The data restore phase performs the actual transfer of data from the dump tape to the 90/30 disc selected. It runs the RESTORE segment of the 9200/9300 DUMP/RESTORE routine under emulation to perform the data transfer. The only significant difference in running the routine on the 90/30 is in the handling of the halt conditions. The 90/30 system requires you to enter replies through system console commands rather than through data switches as was the practice on the 9200/9300 system.

A logical 8410 disc unit is needed for reference in the data cards submitted when using the RESTORE segment of the 9200/9300 DUMP/RESTORE routine. The logical unit number to specify is 14.

A.1.3.4. Disc Extent Identification — Phase 4

Extent identification is the phase that prints out the specific information about the 90/30 pack-file created from the dump tape. It lists the name of the pack-file created and the actual extent of the file now residing on the 90/30 disc. The information printed is derived from the VTOC for the pack-file. In addition, it also prints out specific information for the 8410 files contained in the pack-file. The fields on the printout contain the following information:

- Pack-File Information for 90/30 Disc

  — Pack-file name

    An 8-character name for the file. This name will reside in the FORMAT1 label of the 90/30 disc VTOC. Remember that the entire 8410 disc is only a file on the 90/30 disc.

  — VSN (volume serial number)

  — CRDT (creation date)

    Format is yyddd, where yy represents the year (00—99) and ddd represents the day (001—366).

  — XPDT (expiration date)

    Date when the file may be deleted. The format is the same as the creation date.

  — TYPE (file type)

  — REC. INT (record format)

    The record format is FB (fixed-blocked).

  — BKSZ (block size)

    Size of fixed-length block is 160.

  — RCSZ (record size)

    Size of fixed-length records is 160.

  — EXT CNT (extent count)

    Number of extents for the file will be 1.

  — EXT TYPE (extent type)

— EXT SEQ (extent sequence number)

The extent sequence number will be 1.

— LIMIT (extent limit)

The address specifying the limit of the extent in the form ccchh — ccchh, where ccc is cylinder and hh is head.

- File Information for 8410 Disc

— FILE NAME (file name)

An 8-character name for the 9200/9300 file.

— CRDT (creation date)

Format is yymmdd, where yy represents the year (00—99); mm represents the month (01—12); and dd represents the day (01—31).

— XPDT (expiration date)

Date when the file may be deleted. The format is the same as the creation date.

— GENO (generation number)

Four characters indicating number of this file.

— VOLN (volume number)

Two characters specifying which volume of the file is contained on the disc.

— TYPE (file type)

One of the following five types:

UND	Undefined
ISAM	Indexed sequential
DAM	Direct access
SAM	Sequential
SYSF	SYSFILE

— IND (data set indicator)

Three characters indicating whether this disc is the end of volume or the end of the file for the file.

EOF	End of file
EOV	End of volume

— EXT CNT (extent count)

One to three digits indicating total extent for the volume.

— EXT TYPE (extent type)

One character indicating the type of the extent:

1 Data
2 Overflow area
3 SYSFILE directory

— EXT SEQ (extent sequence)

One to two digits indicating the number of the extent.

— LIMIT (extent limit)

The address specifying the limit of the extent in the form ttss — ttss, where tt is track number and ss is sector number.

A.1.4. Restrictions and Considerations in the Use of PIMAGE Routine

Because the PIMAGE routine preps the entire 90/30 disc area before restoring the 8410 data from the dump tape, you are restricted from using this routine for restoring partial pack-files or single files to a 90/30 pack-file which already contains some data. You may accomplish this, however, by running the DUMP/RESTORE routine as a separate job under emulation just as you would run any 9200/9300 job.

Since most 90/30 discs can contain multiple 8410 pack-files, you should also be aware of the following: If a multiple file 9200/9300 job (i.e., a program using more than one disc file) has all of the files on one physical disc, the head movement involved in executing the job greatly increases the running time.

If a 9200/9300 job required pack changes (two different physical packs assigned to the same physical drive, used during different phases), you must ensure that when the packs are placed on the 90/30 disc drives there will not be a problem in mounting the packs.

To illustrate this problem assume that you have the configuration shown in Figure A—3. In such a configuration, a problem would occur whenever DATAFILE2 is requested. To mount this pack, the pack containing DATAFILE2 would have to be removed. The program, however, still requires DATAFILE1. To resolve this problem, you use an alternate method as shown in Figure A—4. This problem will most likely occur when you have converted to a 90/30 system that has fewer disc units than existed on your 9200/9300 system. Two recommendations are suggested for such users. First, avoid excessive head movement by placing the 8410 data files from different jobs on the same 90/30 disc pack. That is, place files that won't be used in the same job together on a physical pack. Second, when converting to a system having an equal number of disc drives as your previous system, keep the pack setup the same.

You should also be aware that the operating procedure for PIMAGE depends upon whether the OS/3 being used includes spooling. The procedures for these two conditions are as follows:

- Running PIMAGE with spooling

  1. Place the following card deck into the reader:

```
// DATA FILEID=PIMAGEREAD171
   (9200/9300 RESTORE or VTOC control cards)
// FIN
   (PIMAGE control cards or card)
// FIN
```

  2. Enter the following commands from the console:

```
IN
RU PIMAGE
```

- Running PIMAGE without spooling

  1. Place the following card deck into the reader:

```
   (PIMAGE control cards or card)
// FIN
   (9200/9300 RESTORE or VTOC control cards)
// FIN
```

  2. Enter the following command from the console:

```
RU PIMAGE
```

It is required that you mount an input tape when using PIMAGE to create a pack-file. This is not the case, however, when you use PIMAGE to create a workfile. But a logical unit number for the disc must be provided to satisfy the MOS which runs under emulation. This logical unit number of zero is represented as a table displacement of 14. Therefore, your parameter card should be coded as:

```
VTOC 14 xxxxx
```

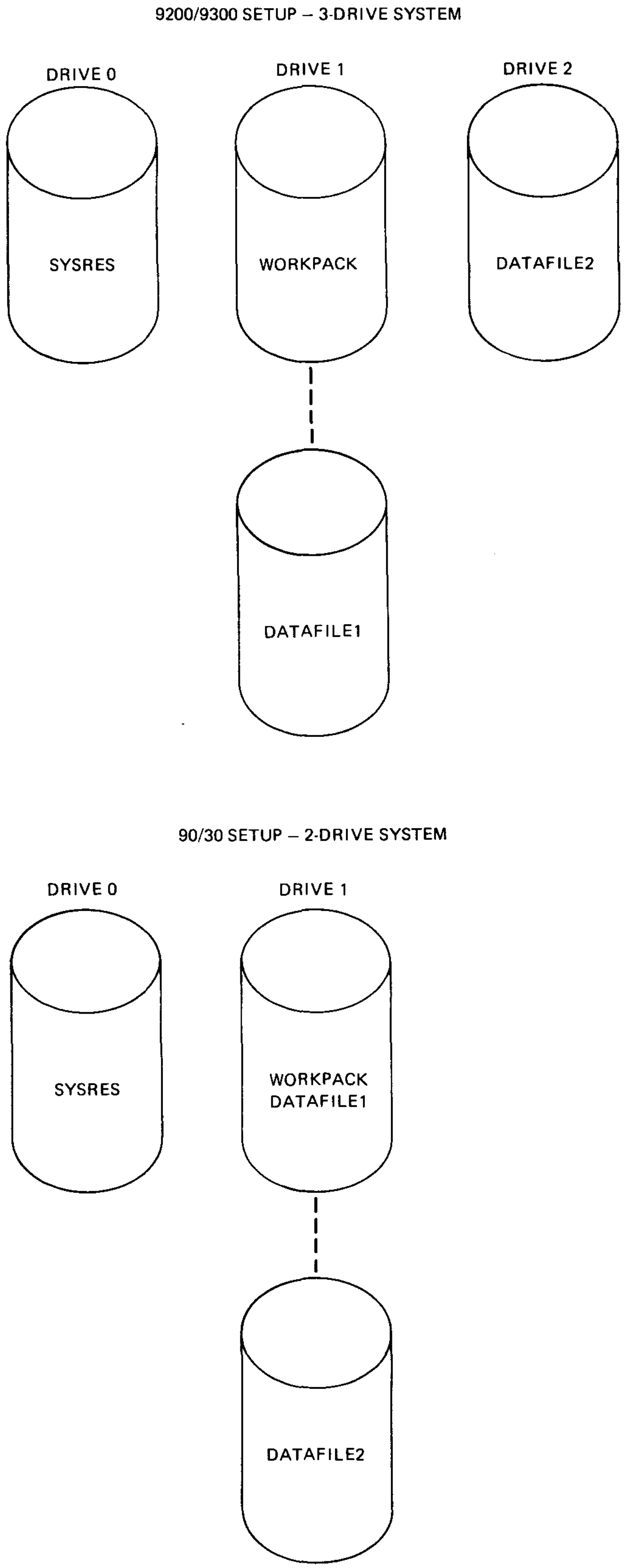


Figure A—3. Multiple-File Configurations

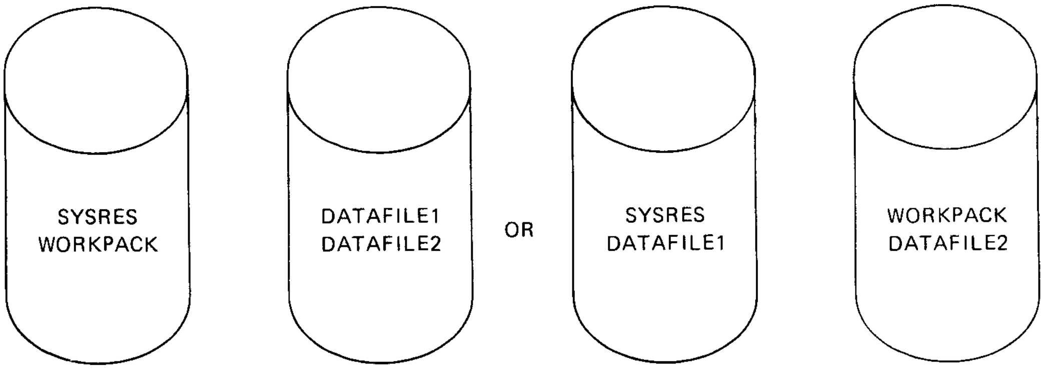


Figure A—4. Alternate Method of Multiple-File Arrangement

A.2. THE 8410 DISC COPYING PROGRAM (COPY$10)

When emulating an 8410 disc, there may be a need to back up the pack—file to another 90/30 disc drive. (ack— file represents all the data files residing on the 8410 disc that is now a file on a SPERRY UNIVAC disc. The COPY$10 program provides an easy method for transferring pack—files, one at a time, from one 90/30 disc drive to another. In addition, COPY$10 can be used for changing the lace factor of pack—files to achieve an optimum disc environment.

A.2.1. Program Initiation

To initiate the COPY$10 program:

1. Determine the placement of the pack-file on a SPERRY UNIVAC disc.

2. Prepare the control cards needed to run the COPY$10 program.

3. Place the control cards in the header and key in RU△COPY$10 from the system console.

A.2.1.1. COPY$10 Control Card Preparation

The conventions for preparation of the COPY$10 control card are:

- The parameters may extend over more than one card.

- The continuation of the parameters onto the second and subsequent cards is indicated by placing an X in column 72 and beginning the second or subsequent card with a // (represents a blank space). The parameters may begin in any column after the //.

- Do not repeat the word COPY$10 on the second or subsequent cards.

- The parameters used on the COPY$10 control card are keyword parameters. A keyword parameter consists of a word or a code immediately followed by an equal sign, which is, in turn, followed by a specification. Keyword parameters can be written in any order in the operand field. Commas are required only to separate parameters.

- Capital letters, commas, equal signs, and parentheses must be coded exactly as shown.

- Lowercase letters and words are generic terms representing information that must be supplied by the user.

- Information contained within braces represents mandatory entires of which one must be chosen.

- Information contained within brackets represents optional entries that (depending upon program requirements) are included or omitted. Braces within brackets signify that one of the specified entries must be chose if that parameter is to be included.

- An optional parameter that has a list of optional entries may have a default specification that is supplied by COPY$10 when the parameter is not specified by the user. Although the default may be specified by the user with no adverse effect, it is considered inefficient to do so. For easy reference, when a default specification occurs in the format delineation, it is printed on a shaded background. If, by parameter omission, COPY$10 performs some complex processing other than parameter insertion, it is explained in the parameter description.

Format:

LABEL	ΔOPERATIONΔ	OPERAND
//	COPY$10	$\left[,\text{L=}\begin{Bmatrix}\text{0}\\ \text{1–9}\end{Bmatrix}\right]$
		$\left[,\text{LI=}\begin{Bmatrix}\text{X}\\ \text{input-label}\end{Bmatrix}\right]$
		$\left[,\text{LO=}\begin{Bmatrix}\text{X}\\ \text{output-label}\end{Bmatrix}\right]$
		$\left[,\text{R=}\begin{Bmatrix}\text{N}\\ \text{Y}\end{Bmatrix}\right]$
		$\left[,\text{VI=}\begin{Bmatrix}\text{X}\\ \text{input-volume-serial-number}\end{Bmatrix}\right]$
		$\left[,\text{VO=}\begin{Bmatrix}\text{X}\\ \text{output-volume-serial-number}\end{Bmatrix}\right]$

L Keyword Parameter:

$L=\left\{\begin{matrix}\mathbf{0}\\ \mathbf{1-9}\end{matrix}\right\}$

Specifies the lacing factor of the output pack-file.

LI Keyword Parameter:

$LI=\left\{\begin{matrix}\mathbf{X}\\ \textbf{input-label}\end{matrix}\right\}$

An alphanumeric entry of up to 44 characters indicating the name of the OS/3 input pack-file.

LO Keyword Parameter:

$LO=\left\{\begin{matrix}\mathbf{X}\\ \textbf{output-label}\end{matrix}\right\}$

An alphanumeric entry of up to 44 characters indicating the name of tthe OS/3 output pack-file.

R Keyword Parameter:

$R=\left\{\begin{matrix}\mathbf{N}\\ \mathbf{Y}\end{matrix}\right\}$

Indicates whether a previous job had allocated an area for a file with the same name as that created by this job.

The file name specified by parameter LO is scratched before the run begins if R=Y.

VI Keyword Parameter:

$VI=\left\{\begin{matrix}\mathbf{X}\\ \textbf{input-volume-serial-number}\end{matrix}\right\}$

A 6-digit alphanumeric entry specifying the volume serial number of the disc containing the input packfile (pack-file to be copied).

VO Keyword Parameter:

$VO=\left\{\begin{matrix}\mathbf{X}\\ \textbf{output-volume-serial-number}\end{matrix}\right\}$

A 6-digit alphanumeric entry specifying the volume serial number of the disc containing the output pack-file (pack-file to be copied).

Example:

```
    LABEL       ΔOPERATIONΔ                  OPERAND                        Δ                  COMMENTS
1                 10       16
// COPY$10 VI=123456,LI=OLDPACK,VO=654321,LO=NEWPACK,L=0
// FIN
```

Copy an 8410 pack-file (OLDPACK) to another 90/30 disc drive (NEWPACK). The volume serial numbers for the discs containing these pack-files are VI and VO, respectively. The lace factor for the output pack-file is established as zero. It is assumed, by default, that no previous job has allocated an area for a file with the same name as the file created by this job.

A.2.1.2. COPY$10 Initiation Through Keyin

The parameter specifications submitted to the COPY$10 program via keyin from the 90/30 system console are exactly the same in both format and content as those for the control card method described in A.2.1.1. The command word format, however, does differ from the operation code specified on the input control card. Therefore, to submit the COPY$10 parameters, key in the command word as shown and duplicate the parameter specifications as indicated in A.2.1.1.

Format:

```
// RUN COPY$10,,
```

A.2.2. Program Restrictions

The COPY$10 program does not have any discernible restrictions.

Appendix B. Emulation 9200 /9300 Operating Instructions

B.1. INITIAL LOADING

Use the spaces provided to insert site-dependent information.

Control Storage Load

- Set DATA ENTRY dials to______.
- Press SYSTEM RESET switch.
- Set INITIAL LOAD CONTROL switch to CONT STOR LOAD.
- Press RUN switch.
- Reset INITIAL LOAD CONTROL switch to OFF.

Initial Program Load

- Set DATA ENTRY dials to______.
- Press SYSTEM RESET switch.
- Set INITIAL LOAD CONTROL switch to PROGRAM LOAD.
- Press RUN switch.
- Reset INITIAL LOAD CONTROL switch to OFF.
- When system console displays:
  IPL TO LOAD STANDARD SUPERVISOR UNLESS NAME KEYED----------

□ Enter__________________________

- Press system console TRANSMIT key.

B.2. EMULATION LOADING

Put these card decks in the card reader:

□ None, job has been filed

□ 90/30 JCL deck

Designation ________________ Location ________________________

□ 9200/9300 Supervisor

Designation ________________ Location ________________________

□ 9200/9300 Job Control Program

Designation ________________ Location ________________________

□ 9200/9300 JCL Deck

Designation ________________ Location ________________________

□ 9200/9300 Program

Designation ________________ Location ________________________

□ Transactions

Designation ________________ Location ________________________

□ Card File

Designation ________________ Location ________________________

B.3. SYSTEM CONSOLE SPECIFICATIONS

Press the system console MESSAGE WAITING key.

□ *Store job in OS/3 job control library.*

- Key in: FILE.
- Press TRANSMIT switch.

□ *Execute job from reader.*

- Key in: RUN.
- Press TRANSMIT switch

□ *Execute job from job control library.*

- Key in: RUN△ ____________________ .
  (Job name)

  (△ Indicates one or more spaces)

- Press TRANSMIT switch.

B.4. EMULATION COMMANDS AND MESSAGES

Operation of the emulator involves commands and messages in three categories. A separate paragraph is devoted to each category:

- B.4.1. Program Directives

  Unsoliciated entries from the operator which direct operation of the emulation program. The emulation program replies are included in this paragraph.

- B.4.2. Display/Alter Commands

  Unsolicited entries from the operator which inspect and modify the program being emulated. The emulation program replies are included in this paragraph.

- B.4.3. Emulator Advisories

  Messages, from the emulator program, which suggest, but do not require, an operator response.

Standard Input Message Procedure

Operator input messages must have at least one (and may have more than one) space where the △ symbol appears in the command format.

Operator input messages may not exceed 20 characters.

The first character entered by the operator, or transmitted to the operator, is the number which the system has assigned to the emulator program.

The second character entered by the operator must be zero.

Input messages immediately follow the second character.

Message variables, shown as lowercase alpha characters (aaaa, xx, yy, etc.), must be entered as hexadecimal values and will be transmitted from the emulator as hexadecimal values unless otherwise noted.

The only input messages which the emulator will accept, *while running*, are STOP, RUN, EOJ, CANCEL, and STATUS. ALL other operator-initiated messages must be preceded by a STOP directive, if the emulator is not then stopped by an error in the emulated program.

To submit a message to the emulator:

- Press system console MESSAGE WAITING key.
- Key in the emulator job number.
- Key in 0 (zero).
- Key in the required directive or display/alter command with its associated parameters, if required.
- Press TRANSMIT switch.

B.4.1. Program Directives

Command Format	Description
CANCEL	The emulator issues a CANCEL to the OS/3 supervisor which terminates the job currently being executed *and any other jobs which were linked together at system generation.* ■ If the remaining jobs are to be executed, use EOJ instead of CANCEL. ■ The emulator will complete any I/O functions then in process, but the emulated program will not regain control to issue I/O commands or close files. *Emulator Condition* To Accept Message: May be stopped or running After Receipt: Running — in the process of terminating *Emulator Reply* None
COREAD R	This directive advises the emulator that one card has been extracted, corrected, and returned to the reader's feed station. All of the cards that followed the corrected card in the reader's output stacker must be returned to the feed station. *Emulation Condition* To Accept Message: May be stopped or running. After receipt: Stopped *Emulator Reply* The emulator will display: 0254 INVALID COMMAND — FIRST OPERAND if the R is missing from the COREAD command.

Command Format	Description
EOJ	The emulator terminates the job currently being executed, but does not cancel any subsequent emulated or OS/3 jobs which were linked to this run at system generation. ■ If the remaining (linked) jobs are to be canceled, use CANCEL. ■ The emulator will complete any I/O functions then in process, but the emulated program will not regain control to issue I/O commands or close files. *Emulator Condition* To Accept Message: May be stopped or running. After Receipt: Running — in the process of terminating
EOJH △ hhhh	The program being emulated is terminated, as described for EOJ, when it attempts execution of an HPR instruction with an operand equal to the hhhh specification. ■ The same effect could be achieved by entering an EOJ directive when the emulator stops to display the HALT message with this same hhhh operand. (See B.4.3.) ■ The emulator will display the HALT message with the specified operand (hhhh), but will not stop if it has been previously given the EOJH directive. If this direction is issued before the HALT message from the emulator, the operator need not respond to that message when it does occur. The EOJ directive does not have to be issued. ■ An equivalent effect may be achieved by specifying an EOJ HALT at system generation. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped *Emulator Reply* ■ None, if hhhh is hexadecimal, or 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful: □ Terminate with □ EOJ □ CANCEL Other ______________________________
HOME	The emulator advances the printer paper to the top line of the next page. See LOOP command for default form sizes. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped *Emulator Reply* None
LOAD	The emulator loads the next program from the card to run under this version of the emulator. The program is then executed, under emulation, without further intervention. ■ This command applies only to card systems (EXEC I, EXEC II, or MOS). *Emulator Condition* To Accept Message: Must be stopped After Receipt: Running *Emulator Reply* None

Command Format	Description
LOAD △ DISC	The emulator loads the next program from disc to run under this version of the emulator. The program is then executed, under emulation, without further intervention. ■ This command applies only to card systems (EXEC I, EXEC II, or MOS). *Emulator Condition* To Accept Message: Must be stopped After Receipt: Running *Emulator Reply* 0611 PROGRAM LOADED — EMULATOR STOPPED
LOOP △ nnnnnn △ pp	The emulator prepares itself to control the printer line advance in accordance with the paper tape loop described as nnnnnn. The name nnnnnn consists of six EBCDIC characters or less, which are assigned (or provided as a default option) at system generation. The printer being emulated is identified by the specification defined in pp; where 00 is the BAR printer, 01 is the 0768 printer, and 02 is the 1004 printer. If no tape loops are stipulated at system generation, the emulator assumes that it controls: — An 11-inch form — At 6 lines/inch — Line 1 = top-of-form — Line 57 = overflow — Line 66 = end-of-flow ■ The emulator assumes that the operator has positioned the printer form to the top line before issuing this tape loop change directive. ■ Initial loading of the first pages tape loop equivalent is automatic, and this command should be used only for subsequent loop mountings for a single execution of the emulator. *Emulator Condition* To Accept Message: May be stopped or running After Receipt: Running *Emulator Reply* ■ If message can be processed: 0256 LOOP nnnnnn IS READY FOR USE ■ If message cannot be processed: 0257 LOOP nnnnnn CANNOT BE LOCATED ■ The tape loop information (a part of emulator generation) must be recompiled and linked to the emulator. or 0258 LOOP nnnnnn NOT DEFINED IN OS/3 JCL STREAM ■ Emulation cannot support any program which utilizes a main storage wrap-around condition because the emulated program simply indexes through 90/30 main storage. *Abnormal Condition Recovery* □ Terminate with □ EOJ □ CANCEL Other ______________________________

Command Format	Description
MOUNT△x△yy	Directs the emulator to a 9200/9300 pack-file. This is equivalent to mounting an 8410 pack on the 9200/9300. The pack-file may be resident on one of the currently mounted 90/30 packs or on a 90/30 pack to be mounted. This directive is normally entered when the emulator has stopped as a result of the emulated program HPR. x = the 9200/9300 physical unit number of the drive which was used to mount the pack in question. The entry is a numeric value from 1 to 4. yy = nn, where nn represents a 2-digit mount number specified at system generation. May be in the range 01—99. or yy = NX, where NX is a literal which directs mounting of the next pack described during system generation. yy = DS, where DS is a literal which displays the current mount number; MNT NN is displayed on console if disc is mounted. CLOSED is displayed if disc is not mounted. If yy is either nn or NX, the emulator will ask OS/3 to direct mounting of a new 90/30 disc pack unless the 9200/9300 files are on the 90/30 pack being read. yy = CL, where CL is a literal which directs that current files, comprising a single 9200/9300 disc pack, be prematurely closed. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (Alteration is made during stop condition.) *Emulator Reply* ■ If the value x is out of range or if a required 90/30 disc drive is not available: 0254 INVALID COMMAND — FIRST OPERAND ■ If the mount number is invalid, if the files specified are the last 9200/9300 disc pack recognized, or if the specified files cannot be located: 0255 INVALID COMMAND — SECOND OPERAND
OP △ REQ △ xx	The emulator performs the I/O and interrupt functions which correspond with the xx operand of the OP REQ specification. ■ The OP REQ directive is the functional equivalent of setting a value into the DATA ENTRY switches on the 9200/9300 console and pressing the OPREQ switch. *Emulator Condition* To Accept Message: Must be either stopped or running After Receipt: Running *Emulator Reply* None, if OP REQ can be processed ■ If message cannot be processed: 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) or 0499 OP REQUEST INHIBITED — COMMAND NOT EXECUTED *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________

Command Format	Description
RUN	Restarts the emulator *at the point it previously stopped.* If the previous stop was the result of an error condition, that error condition will be repeated and the emulator will once again stop, unless the condition is corrected. The emulator stops on the instruction which caused the error condition. ■ The directive is the functional equivalent of pressing RUN on the control panel of the 9200/9300. *Emulator Condition* To Accept Message: Either stopped or running After Receipt: Running. *Emulator Reply* None
STATUS	The emulator reports its current condition. *Emulator Condition* To Accept Message: Either stopped or running After Receipt: Unchanged *Emulator Reply* ■ Normal condition 0340 OPERATING or 0342 STOPPED BY OPERATOR COMMAND ■ Abnormal Condition 0341 STOPPED — AWAITING RESPONSE TO HALT 0343 STOPPED — ADDRESS ERROR OUTSTANDING 0344 STOPPED — DATA EXCEPTION OUTSTANDING 0345 STOPPED — DIVIDE EXCEPTION OUTSTANDING 0346 STOPPED — INVALID OP CODE OUTSTANDING 0347 STOPPED — SPECIFICATION ERROR OUTSTANDING *Abnormal Condition Recovery* □ Terminate with □ EOJ □ CANCEL Correct condition by ______________________________ ______________________________
STOP	The emulator goes into a stop condition. ■ The directive is the functional equivalent of pressing STOP on the 9200/9300 control panel ■ Any error of the emulated program puts the emulator into a stop condition. If an error occurs in the emulator, itself, the emulator will terminate unconditionally. ■ If a 90/30 system error occurs, the operator will be notified, but the emulator will remain in a run condition. The operator must submit a STOP command before any command requiring a stop state can be entered. *Emulator Condition* To Accept Message: Either stopped or running After Receipt: Stopped *Emulator Reply* None

B.4.2. Display/Alter Commands

Command Format	Description
ALTRIC △ yyyyyyyy	The emulator replaces the entire I/O program state control word of the emulated program with the value yyyyyyyy. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* The emulator accepts a valid operator specification with one of two replies: ■ If the IPSC which is to be altered is valid: 0269 IPSC ALTERED FROM xxxxxxxx TO yyyyyyyy ■ If the IPSC which is to be altered was, itself, invalid: 0269 IPSC ALTERED FROM INVALID TO yyyyyyyy where: xxxxxxxx = content before alteration yyyyyyyy = content after alteration The operator rejects an invalid operator specification with one of four replies: 0264 INVALID — ADDRESS IS BEYOND STORAGE LIMIT 0265 INVALID — ADDRESS IS BELOW 64 0266 INSTRUCTION ADDRESS IS NOT ON HALFWORD BOUNDARY 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________
ALTRM △ aaaa △ yy	The emulator replaces the current content of the emulated program location aaaa with yy. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* ■ If message can be processed: 0260 ALTERED STORAGE LOCATION aaaa FROM xx TO yy where: aaaa = storage address xx = location content before alteration yy = location content after alteration ■ If the message cannot be processed: 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) or 0264 INVALID ADDRESS IS BEYOND STORAGE LIMIT *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other______________________________

Command Format	Description
ALTRIP △ yyyy	The emulator substitutes the value, yyyy, for the address of the next instruction in the emulated program I/O program state control word. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition). *Emulator Reply* The emulator accepts a valid operator specification with one of two replies: ■ If the IPSC instruction address which is to be altered is valid: 0270 IPSC INSTRUCTION ADDRESS ALTERED FROM aaaa TO yyyy ■ If the IPSC instruction address which is to be altered was, itself, invalid: 0270 IPSC INSTRUCTION ADDRESS ALTERED FROM INVALID TO yyyy where: aaaa = instruction address before alteration yyyy = instruction address after alteration The operator rejects an invalid operator specification with one of four replies: 0264 INVALID — ADDRESS IS BEYOND STORAGE LIMIT 0265 INVALID — ADDRESS IS BELOW 64 0266 INSTRUCTION ADDRESS IS NOT ON HALFWORD BOUNDARY 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________
ALTRIR △ r △ yyyy	The emulator replaces the current content of the emulated program's I/O register r with the value yyyy. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* ■ If message can be processed: 0263 I REGISTER N ALTERED FROM xxxx TO yyyy where: r = register number (8 through F) xxxx = content before alteration yyyy = content after alteration ■ If the command cannot be processed: 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) or 0500 INVALID ENTRY — REGISTER MUST BE 8 — F *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________

Command Format	Description
ALTRPC △ yyyyyyyy	The emulator replaces the entire processor program state control word of the emulated program with the value yyyyyyyy. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* The emulator accepts a valid operator specification with one of two replies: ■ If the PPSC which is to be altered is valid: 0269 PPSC ALTERED FROM xxxxxxxx TO yyyyyyyy ■ If the PPSC which is to be altered was, itself, invalid: 0269 PPSC ALTERED FROM INVALID TO yyyyyyyy where: xxxxxxxx = content before alteration yyyyyyyy = content after alteration The operator rejects an invalid operator specification with one of four replies: 0264 INVALID — ADDRESS IS BEYOND STORAGE LIMIT 0265 INVALID — ADDRESS IS BELOW 64 0266 INSTRUCTION ADDRESS IS NOT ON HALFWORD BOUNDARY 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________
ALTRPP △ yyyy	The emulator substitutes the value yyyy for the address of the next instruction in the emulated program processor program state control word. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* The emulator accepts a valid operator specification with one of two replies: ■ If the PPSC instruction address which is to be altered is valid: 0270 PPSC INSTRUCTION ADDRESS ALTERED FROM aaaa TO yyyy ■ If the PPSC instruction address which is to be altered was, itself, invalid: 0270 PPSC INSTRUCTION ADDRESS ALTERED FROM INVALID TO yyyy where: aaaa = instruction address before alteration yyyy = instruction address after alteration The operator rejects an invalid operator specification with one of four replies: 0264 INVALID — ADDRESS IS BEYOND STORAGE LIMIT 0265 INVALID — ADDRESS IS BELOW 64 0266 INSTRUCTION ADDRESS IS NOT ON HALFWORD BOUNDARY 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________

Command Format	Description
ALTRPR △ r △ yyyy	The emulator replaces the current content of the emulated program processor register r with the value yyyy. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (alteration is made during stop condition) *Emulator Reply* ■ If message can be processed: 0263 REGISTER r ALTERED FROM xxxx TO yyyy where: r = register number (8 through F) xxxx = content before alteration yyyy = content after alteration ■ If the command cannot be processed: 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) or 0500 INVALID ENTRY — REGISTER MUST BE 8 — F *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ______________________________
DISPIC	The emulator displays the content of the emulated program I/O program state control word. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (display is made during stop condition) *Emulator Reply* ■ If the content is valid, or if the instruction portion is wrong but within the emulated program assigned area: 0268 CURRENT IPSC CONTENT xxxxxxxx where: xxxxxxxx = PSC content ■ If the PSC instruction portion had been set to an invalid address (one which is out of the emulated program assigned area), the emulator will stop and display ADDRESS ERROR. If this command is then executed, the reply would be: 0268 CURRENT IPSC CONTENT INVALID *Abnormal Condition Recovery* □ Terminate with □ EOJ □ CANCEL Other ______________________________

Command Format	Description
DISPIR △ r	The emulator displays the content of the emulated program I/O register r. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (display is made during stop condition) *Emulator Reply* ■ If message can be processed: 0262 I REGISTER r CONTAINS xxxx where: r = register number (8 through F) xxxx = register content ■ If the register specification is not 8 through F: 0500 INVALID ENTRY — REGISTER MUST BE 8 — F *Abnormal Condition Recovery* Correct register specification. If impossible: □ Terminate with □ EOJ □ CANCEL Other ____________________
DISPM △ aaaa	The emulator displays the content of the emulated program location aaaa. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (display is made during stop condition) *Emulator Reply* ■ If message can be processed: 0259 STORAGE LOCATION aaaa CONTAINS xx where: aaaa = emulated program address xx = content ■ If message cannot be processed: 0267 INPUT MESSAGE CONTAINS NON-HEXADECIMAL CHARACTER(S) or 0264 INVALID — ADDRESS IS BEYOND STORAGE LIMIT *Abnormal Condition Recovery* Correct entry and retry. If unsuccessful or impossible: □ Terminate with □ EOJ □ CANCEL Other ____________________

Command Format	Description
DISPPC	The emulator displays the content of the emulated program's processor program state control word. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (display is made during stop condition) *Emulator Reply* ■ If the content is valid, or if the instruction portion is wrong but within the emulated program assigned area: 0268 CURRENT PPSC CONTENT xxxxxxxx where: xxxxxxxx = PSC content ■ If the PSC instruction portion had been set to an invalid address (one which is out of the emulated program assigned area), the emulator will stop and display ADDRESS ERROR. If this command is then executed, the reply would be: 0268 CURRENT PPSC CONTENT INVALID *Abnormal Condition Recovery* □ Terminate with □ EOJ □ CANCEL Other ________________
DISPPR △ r	The emulator displays the content of the emulated program's processor register r. *Emulator Condition* To Accept Message: Must be stopped After Receipt: Stopped (display is made during stop condition) *Emulator Reply* ■ If message can be processed: 0262 I REGISTER r CONTAINS xxxx where: r = register number (8 through F) xxxx = register content ■ If the register specification is not 8 through F: 0500 INVALID ENTRY — REGISTER MUST BE 8 — F *Abnormal Condition Recovery* Correct register specification. If impossible: □ Terminate with □ EOJ □ CANCEL Other ________________

B.4.3. Emulator Advisories

The emulator advisories (system consle messages) are in the system messages programmer/operator reference, UP-8076 (current version).

B.4.4. Input Message Summary

B.4.4.1. Program Directives

Directive	Description
CANCEL	Terminate current job step and all subsequent job steps
COREAD R	Read corrected card from reader feed station
EOJ	Terminate current job step only
EOJH △ hhhh	Terminate, as in EOJ, when HPR hhhh is detected
HOME	Emulator to home (eject) form
LOAD △ (DISC)	Load job from card reader (or disc)
LOOP △ nnnnnn	Operate in accordance with new printer loop
MOUNT△x△yy	Mount pack identified
OP REQ △ xx	Perform operator request function xx
RUN	Restart emulator
STATUS	Determine emulator condition
STOP	Stop emulator

B.4.4.2. Display/Alter Commands

Command	Description
ALTRIC △ yyyyyyyy	Change entire I/O PSC
ALTRIP △ yyyy	Change I/O PSC instruction address
ALTRM	Change storage location
ALTRPC △ yyyyyyyy	Change entire processor PSC
ALTRPP △ yyyy	Change processor PSC instruction address
ALTRIR △ r △ yyyy	Change I/O register
ALTRPR △ r △ yyyy	Change processor register
DISPIC	Display IPSC
DISPM	Display storage
DISPPC	Display PPSC
DISPPR △ r	Display processor register
DISPIR △ r	Display I/O register

B.5. EMULATOR ERROR CODES

The emulator generates certain error codes when an unrecoverable error occurs. These error codes are passed to the supervisor and ultimately they appear in the error code field of a cancellation dump. You can find these error codes in the system messages programmer/operator reference, UP-8076 (current version).

Appendix C. Conversion Utilities

C.1. DISC FILE CONVERSION – SPERRY UNIVAC 9200/9300 TO SPERRY UNIVAC 90/30 VIA OS/3 UNLOAD/RELOAD CONVERSION UTILITY

As a SPERRY UNIVAC 9200/9300 customer converting to the SPERRY UNIVAC 90/30 System, you are faced with the problem of file conversion. That is, the files contained on your 8410 discs cannot be run on the 90/30 system since this particular disc is not supported by either the hardware or software of the 90/30 system. In addition, the files contained on your 8411 or 8414 discs also must be converted since these files are not acceptable to the OS/3 data management due to differences in the volume table of contents (VTOC) and, in most cases, the data format and conventions. Therefore, a conversion utility capable of transferring your present files to a 90/30 disc is needed. This utility is provided to you as a 2-program package referred to as the SPERRY UNIVAC Operating System/3 (OS/3) UNLOAD/RELOAD disc file conversion utility. See Figure C—1.

The UNLOAD program is used to unload your existing data files to magnetic tape or 80-column punch cards which are used as input to the RELOAD program. RELOAD uses this input to re-create your files on any 90/30-supported disc subsystem.

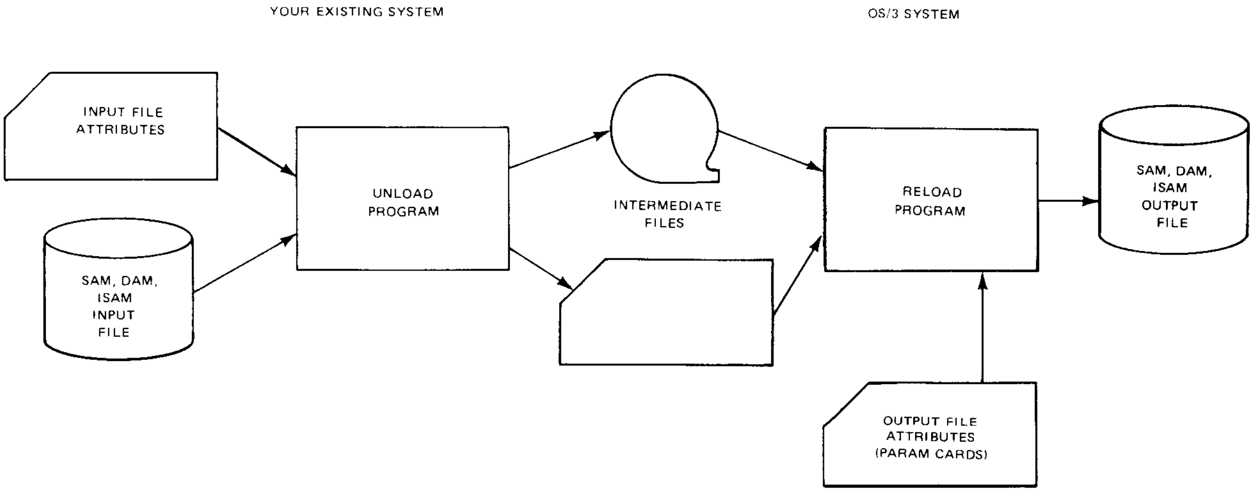


Figure C—1. UNLOAD/RELOAD Disc File Conversion Utility

The UNLOAD program is also used to copy your existing data files to magnetic tape or 8411/8414 disc SAM files (Figure C—2). These files are directly usable by your applications programs or by the data utility on the 90/30 system.

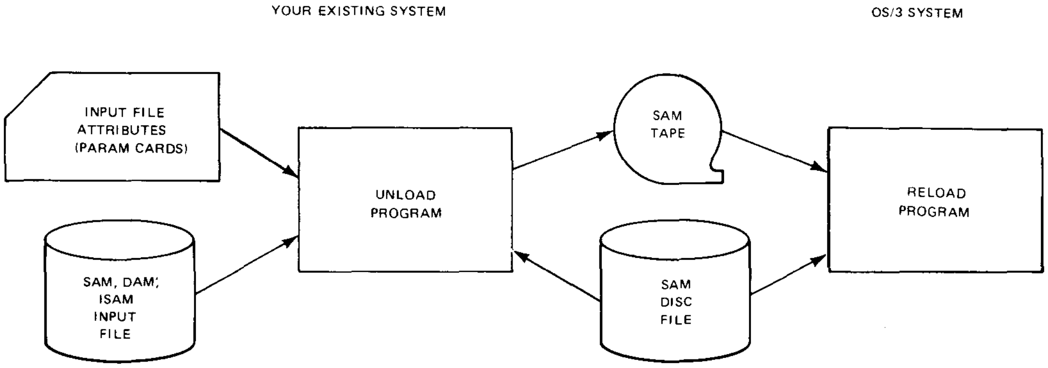


Figure C—2. UNLOAD to SAM File Conversion

Because the files of two different systems are involved in file conversion, confusion over the use of terms such as *input files, intermediate files,* and *output files* may arise. To avoid such confusion, the following definitions apply: *Input files* refer to the original files of your existing system. They are the source from which the UNLOAD program produces an *intermediate file* on 80-column punched cards or on magnetic tape, or an output file on 8411/8414 disc or on magnetic tape. The intermediate file serves as the input to the RELOAD program, which transcribes or re-creates these files as an OS/3 formatted file on a 90/30 disc. The new file produced is called the *output file.*

C.1.1. Disc File UNLOAD Program

The disc file UNLOAD program is supplied as part of the current OS/3 software release and is executed under control of the current 9200/9300 DNCOS, either on the 9200/9300 system or on the 90/30 system under emulation. There are three versions of the UNLOAD program:

- UNLOAD for 8411/8414 disc input
- UNLOAD10 for 8410 disc input
- UNLD1011 for 8410 disc input plus 8411/8414 disc output

You are given the option of linking any version of UNLOAD. Although three versions of the UNLOAD program are supplied, they will be collectively referred to in this section as UNLOAD unless otherwise noted.

C.1.1.1. Program Objectives

The primary objectives and capabilities designed into the UNLOAD program can be summarized as follows:

1. Conversion of as many files as desired in a single execution of the program.

2. User control of the types of input and output files via specification cards inserted into the control stream used to execute UNLOAD.

3. Execution under your current DNCOS supervisor or, in the case of the UNLOAD10 version, optionally executable under a tape resident supervisor.

4. Capable of being run on your 9200/9300 system prior to the 90/30 installation or on the 90/30 system under emulation. In the latter case, the 8410 discs cannot be used since they are not compatible with 90/30 hardware.

5. Main storage conservation by designing the loadable version of this program as a number of overlays that are fetched into main storage only when needed by the current function being performed.

6. Provides a comprehensive printer log that includes information needed to identify the files transcribed, label information, and record counts; also provides diagnostic messages via halt displays and printer log that advise you of any error conditions which are encountered during program execution.

7. Provides you with a printer output to the standard bar printer and a card output, if desired, to any one of three punches supported by the 9200/9300 system (serial, row, or 1004 punches).

C.1.1.2. Interfacing With UNLOAD

The conventions used in the UNLOAD program are those of the 9200/9300 standard IOCS for disc input and output and for unit record input and output. The output to tape is accomplished via SRCs issued to the supervisor. Some disc I/O (principally reading and writing of VTOC information) will also be done via SRCs.

The data base is that of the 9200/9300 system. All disc input files must be on 8410, 8411, or 8414 disc packs in any format supported by 9200/9300 disc IOCS; namely, sequential (SAM), direct access (DAM), or indexed sequential (ISAM). The only restriction imposed is that no DAM files other than relative-record-addressable files can be read.

The operator interfaces with UNLOAD program through three types of printer displays:

- Input statement listing; your specification card appears in print positions 11 through 90 and is flagged by dollar signs ($) in print positions 1 through 8.

- Information printouts of material, such as record counts, end-of-volume, end-of-file, end-of-job notification, format of output data, and label information (input and output files). Information printouts are identified by asterisks (*) appearing in print positions 1 through 8 of their first line.

- Error messages including descriptive printouts accompanied by halt displays concerning error conditions encountered during the execution of UNLOAD. All halt displays are identified by exclamation points (!) in print positions 1 through 8.

C.1.1.3. Input Requirements to UNLOAD

There are two types of input to the UNLOAD program: the disc files to be copied and the card files describing the disc input.

The disc input to UNLOAD may be sequential (SAM), direct (DAM), or indexed sequential (ISAM) files. UNLOAD transcribes the entire contents of the file to the output medium.

For SAM and ISAM files, any record format currently acceptable to 9200/9300 IOCS is accepted by UNLOAD. For DAM, the file is read by the 9200/9300 SAM and must be relative-record-addressable. For DAM, the user has the option of ignoring, copying, or terminating on records with contents of all binary 0's. If SAM or DAM files contain key fields, such keys will be ignored.

The card input to UNLOAD may be divided into two types: PARAM cards, which are processed by the supervisor during the loading of UNLOAD, and specification cards, which are read by UNLOAD during program execution.

In the absence of information inserted by PARAM cards, UNLOAD assumes that it is to produce tape or disc output files, as determined by file specification cards, which are directly usable by your programs or the OS/3 data utilities.

If the following PARAM card is used:

```
/ PARAM 0001,01,1
```

tape or card intermediate files, as determined by the file specification cards, are produced and are acceptable only to the RELOAD program.

One other set of options may be invoked through the use of PARAM cards. These options pertain to the conversion of creation and expiration dates from the 9200/9300 input file to the OS/3 Julian date format on the intermediate or output media. Since 9200/9300 users are not forced to use any particular date format, the following PARAM card may be used to tell UNLOAD which format dates are on the 9200/9300 disc files:

```
/ PARAM 0005,01,2 if mmddyy

/ PARAM 0005,01,3 if ddmmyy

/ PARAM 0005,01,4 if yyddd
```

If none of these PARAMs is used, UNLOAD assumes that 9200/9300 dates are in the yymmdd format. If a date field contains blanks, 0's, or all bits (DATE card omitted), the converted date will be all 0's.

Any PARAM cards that are used when executing UNLOAD must appear between the EXEC statement and the DATA statement.

The file specification cards describe the characteristics of the input to and the output from files produced by UNLOAD. These cards are identified by the character F that appears in column 1 of the card. One file card is required for each input file to be copied to the intermediate or output file. The format of the file specification card is presented in Table C—1.

In addition to the file specification cards, there are halt specification cards. The halt specification card is identified by the character H in card column 1 and is used to halt processing so that the operator may perform duties such as mounting a new disc pack or to check on the progress of the UNLOAD program. The format of the halt specification card is presented in Table C—2.

Table C—1. File Specification Card Format (Part 1 of 3)

Card Column	Value	Description
1	F	Identifies card type
2—3	XX	The input logical unit number, in hexadecimal
4—5	XX	The output logical unit number, in hexadecimal
6—11	XXXXXX	Output volume serial number; if output is to tape, this VSN will be written on the tape. If output is to disc, UNLOAD will verify that this is the VSN of the disc pack which is mounted on the output logical unit. When output is to tape, UNLOAD ensures that the tape VSN begins with an alphabetic character and consists of all alphanumeric characters, as required by OS/3 data management. When multivolume tape output is produced, UNLOAD increments the last character of the output VSN by 1 each time a new output volume is started. When "stacking" files on a tape, the VSN used for the first file in the group is carried forward for all subsequent tape output files until it is necessary to rewind a tape with interlock and begin a new output group. For multivolume disc output, the VSN of the disc pack on which a file began appears in the FORMAT-1 label of each subsequent volume of the file. When each new disc pack is mounted, UNLOAD prints its volume serial number and halts (MSG 7726) to permit the operator to verify that the correct pack is mounted. Discs and tapes which are to be used under OS/3 should have unique VSNs.
12	I	Specified only if output is to tape which is to be rewound with interlock upon completion. Unless otherwise directed by this specification, UNLOAD leaves an output tape extended until such time as a new tape output unit is specified or until it reads a sentinel card (/*), at which time it rewinds with interlock the last output tape in use.

NOTE:

A blank occurring in a decimal field will be treated as a zero.

Table C—1. File Specification Card Format (Part 2 of 3)

Card Column	Value	Description
13—14	FU FB VU VB UN	Record format of the input file. Output in RELOAD format is always in variable-length blocked format. Other output is the same format as input, except for undefined, which is converted to variable-length unblocked. When specifying the record format of 8411/8414 ISAM files, you should remember that files created by RPG programs are always considered to be blocked files, even though they may be blocked one to one. This causes, on the 9300, the use of an embedded key field in addition to the external hardware-oriented key field. If FU is specified, UNLOAD must append the external key field to the front of each record and increase the block size on output. If FB is specified, the redundant external key field required by 9300 RPG ISAM processing will not be appended. "FU" should probably be specified only for files which were created and processed by non-RPG programs. If the file organization is direct, record format is assumed to be fixed-length unblocked, and either IG, EN, or blanks may be punched in these columns, to indicate the selection of one of the following options: IG Ignore, i.e., do not copy records consisting of all binary 0's. EN Cease copying the file if a record is encountered which consists of all binary 0's. Blanks Copy the entire file until end-of-file detected.
15	3	Specified only if the input file is an 8410 ISAM file which was created by an assembler language program using the optional 9300 stagger effect (PROC=9300 specified in DTFIA).
16—19	nnnn	Record size in decimal
21—24	nnnn	Block size in decimal; due to the peculiarities of the respective 9300 IOCS routines, when transcribing a fixed-length blocked file, you may specify a larger block size than really exists in the input file in order to cause reblocking of the output file. The exception to this rule is 8411 ISAM, for which you must specify the true input block size. NOTE: Users of 8410 must also specify block size as a multiple of record size even though you may have been required to specify 160 when compiling RPG programs which processed the files.
26—28	nnn	For ISAM only, key length in decimal
30—33	nnnn	For ISAM only, key location relative to the beginning of record, with 0000 signifying the first byte of the record

NOTE:

A blank occurring in a decimal field will be treated as a zero.

Table C—1. File Specification Card Format (Part 3 of 3)

Card Column	Value	Description
35	$\left\{\begin{matrix} I \\ S \\ D \end{matrix}\right\}$	File organization
36	Blank or 1—8	For ISAM only, the number of volumes comprising the file, with blank assumed to mean 1. All volumes must be online while executing UNLOAD. UNLOAD assumes that volumes of an ISAM file will be mounted on consecutive logical units.
37—80	$X_1\ X_2 \ldots X_{44}$	File identification, i.e., the 44-byte (8 for 8410) character string which uniquely identifies the file to be copied. The first 17 characters of this field are used to create the file ID in the HDR1 label when output is to tape. Although only the first eight characters are used for locating an 8410 input file, additional characters may be appended to expand the file ID on the output medium. When UNLOAD writes HDR1 or FORMAT-1 labels, it converts the following characters in the file-id to blanks: + - /(

NOTE:

A blank occurring in a decimal field will be treated as a zero.

Table C—2. Halt Specification Card Format

Card Column	Value	Description
1	H	Identifies card type
3—6	XXXX	4-digit hexadecimal value to be displayed on system console. Must not be greater than 7FFF
8—80	$X_1X_2 \ldots X_{73}$	Instructions to operator to be displayed on the printer

C.1.1.4. Output Files Produced by UNLOAD

The OUTPUT produced by the UNLOAD program consists of disc output, tape output, card output, and printer output.

C.1.1.4.1. Disc Output

The disc output from UNLOAD consists of SAM files which are acceptable to OS/3 data management. The output file has the same record size as the input file. Its block size is as specified in the F card. Since undefined format is not supported by OS/3 SAM, undefined input produces variable-length, unblocked output. Because these disc files are written by the standard 9200/9300 SAM processor, UNLOAD allocates all remaining pack space to the file on its 9200/9300-compatible FORMAT-1 label before each file is opened. After the file is closed, UNLOAD deallocates unused space and generates an OS/3-compatible Format-2 label and such other information as is necessary within the Format-1 label to make the file acceptable to OS/3 data management. Once these modifications have been made to a pack's VTOC, the pack should be considered unusable by any 9200/9300 program but UNLOAD.

Prior to running UNLOAD to produce the first OS/3-compatible files on a pack, you should use the direct access space management program (DASM) to deallocate all files and write a fresh VOL1 label, preferably with a unique volume serial number. If you need to wipe out an UNLOAD-created VTOC in order to reuse a disc pack for UNLOAD or for normal 9200/9300 files, it is necessary to reprep the pack and then reDASM it.

UNLOAD uses the ownership field of the VOL1 label to indicate whether it has converted a VTOC to OS/3 format. If an output pack has not yet had its VTOC converted to OS/3 format, UNLOAD zero fills all Format-1 labels to clear any previous file allocations and updates the Format-5 label accordingly. It updates the VOL1 label with the characters OS/3 DATA in the ownership field. (Note that UNLOAD requires that the VTOC of the output pack be on cylinder 0).

In a 2-drive configuration, it is necessary to have a TRANSCYL and minimal SYSFILE on the input pack to permit UNLOAD to execute. Since most user data packs contain scratch files for sorting, it would probably be a simple matter to scratch one such file to make space available for allocation to SYSFILE and TRANSCYL. On an 8411 disc, SYSFILE and TRANSCYL would consume about four cylinders; on an 8414, two cylinders. The following example is a sample 8411 run stream for transferring UNLOAD and minimum system elements to a data pack on logical unit 1 to permit operation in a 2-drive environment. This example uses 111111 for the VSN of the system pack on drive 0, and 222222 for the VSN of the data pack; you should substitute whatever VSNs actually exist on the packs used.

Example:

```
 LABEL   ΔOPERATIONΔ         OPERAND               Δ            COMMENTS
1        10    16
/        JOB   PREPARE 9300 PACK FOR USE AS INPUT TO "UNLOAD"
/        NOTE  IN A TWO-DISC SYSTEM
/        EXEC  DASM
/        DATA  T
 DVC 01
 SCR 222222,SCRATCH1
 LBL TRANSCYL,222222,01,SQ
 EXT C,00,CYL,0002                    * 0001 IF 8414
 LBL SYSFILE,222222,01,SQ
 EXT C,00,TRK,0001
 EXT C,00,CYL,0002                    * 0001 IF 8414
 END
/*
/        EXEC  LIBS
/        DATA  C
/ CTL    N,W,1,SYSFILE,R,O,SYSFILE
/ BOOT   CLEAR
/ SELECT /SUPR
/ SELECT /JBCN
/ SELECT /UNLOAD
/*
/        EXEC  DPRS
/        DATA  C
 DUMP D 00 111111 TRANSCYL
 UNIT D 01 TRANSCYL                                                      222222
/*
/        FINIS
```

C.1.1.4.2. Tape Output

The tape output from UNLOAD will always consist of tapes which are acceptable to OS/3 data management. That is, they will have VOL1, HDR1, and HDR2 labels and appropriate EOV/EOR labels and tapemarks (Figure C—3). Such tapes may be multifile volumes and/or multireel files.

If the tapes are destined for input to the 90/30 RELOAD program, the data in each file will be preceded by a truncated data block containing two directory records which are required by RELOAD. Format of the two directory records is presented in Tables C—3 and C—4, respectively.

All data on a RELOAD format tape, including the directory records, is in variable-length format. For tapes which will be read by RELOAD, block size is determined as follows:

Let B be input block size and T be maximum tape block size:

$$B_1 = \begin{cases} \text{B for variable-length input records} \\ \text{B+8 for fixed and undefined records} \end{cases}$$

T= Maximum (2048 or B_1). If available main storage is limited, maximum may be decreased below 2048.

If the tape output file is destined for unrestricted use on the 90/30 system, its record format and block size will be identical to that of the disc input file, again with the restriction that undefined format will be converted to variable-length unblocked format.

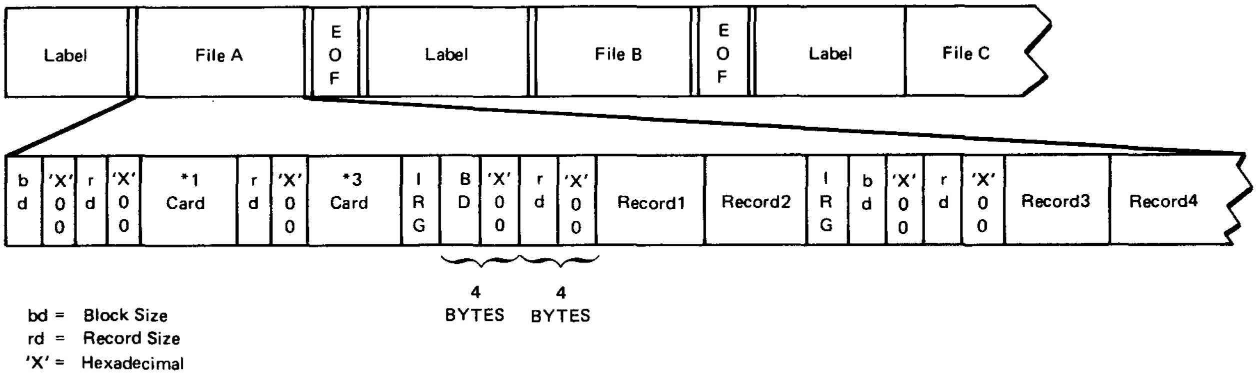


Figure C—3. Format of Output Tape Generated by UNLOAD

Table C—3. First File Directory Record Format (Part 1 of 2)

Card Column	Value	Description
1—2	*1	Identifies card as a directory restore card.
3—46	$I \ldots I_{44}$	File identification — the 44-character file identification placed in the file label which uniquely identifies the file that was dumped.
47—52	Blank	Not used
53—55	nnn	The creation date — the day and year the file was created in discontinuous binary. Each byte is generated by converting two digits of the 9200/9300 data field to binary.
56—58		Not used

Table C—3. First File Directory Record Format (Part 2 of 2)

Card Column	Value	Description
59—61	nnn	Expiration date — the day and year the file may be deleted. Generated in same way as creation date.
62—64	Blank	Not used
65	U	Code indicates source machine is 9200/9300.
66	I S D	Indicates file organization where: I indicates indexed sequential (ISAM). S indicates sequential (SAM). D indicates direct access (DAM).
67—80	Blank	Reserved for expansion

Table C—4. Second File Directory Record Format

Card Column	Value	Description
1—2	*3	Identifies card type as a directory restore card.
3—4	FB FU VU VB UN	Record format blocking factor where: V represents variable-length record. F represents fixed-length record. B represents blocked record. U represents unblocked records. UN represents undefined.
5—9	nnnnn	Record length — the length of a record in number of bytes
10—14	nnnnn	Block length — the length of a block in number of bytes
15—17	nnn	Key length — the length of an ISAM record key in number of bytes
18—22	nnnnn	Key location — the location of the record key within the record, relative to the first position of the record, which would be location 00000.
23—80	Blank	Reserved for future expansion

C.1.1.4.3. Card Output

The card output from UNLOAD consists of EBCDIC card images in the format required by the RELOAD program. Each file is preceded by two directory records; as many cards are punched as are required to hold each logical disc record, including RDW fields.

Each new record starts on a new card. The format of the output cards is listed in Table C—5. All of the cards have the same format with the exception of the first card of each logical record in which columns 1 and 2 contain the length of data and columns 3 and 4 contain binary 0's (00); all remaining cards contain user information in these fields.

Table C—5. Output Card Format

Card Column	Value	Description
1—2*	nn	Length of data (in binary) that follows on 1—n cards
3—4*	00	Binary 0's
5—74	User data	First portion of user record
75	Blank	Reserved for future use
76—80	nnnnn	Card sequence number in hexadecimal

*Applicable only to the first card of each record; all remaining cards contain user information in these fields.

C.1.1.4.4. Printer Output

The printer output from UNLOAD consists of a log of information concerning those files which were copied, including such information as was provided on input specification cards as well as information derived from disc VTOC labels and counts of the number of records copied for each file. Other information such as error messages and user-supplied H card messages is also included in the printer file.

C.1.1.5. Preparing the UNLOAD Program for Use

The modules required for generation of the UNLOAD program resides in the OS/3 source library format as four elements:

- CU$3REL — which is a 9300 librarian run stream that includes all necessary relocatable elements. It copies the UNLOAD relocatables to a disc library file named SCRATCH1 on logical unit 0.

- CU$3L10 — a 9200/9300 linker run stream for linking UNLOAD10 and adding it to SYSFILE on logical unit 0

- CU$3L11 — A 9200/9300 linker run stream for linking UNLOAD

- CU$3L12 — A 9200/9300 linker run stream for linking UNLD1011

  Also supplied with the OS/3 release is a "canned" OS/3 run stream named CU$PUNCH which may be used to assist you in punching out the desired combination of the pseudo-source decks previously named. To run CU$PUNCH, type in RUN CU$PUNCH at the 90/30 system console after inserting the following cards in the system reader:

```
//CU$T△JSET△n

//△FIN
```

If the value following JSET is 4, CU$3L10, followed by CU$REL, will be punched; if 5, CU$3L11, followed by CU$3REL; if 6, CU$3L12 followed by CU$3REL. In each case, the first of the two decks is a very short deck, consisting mostly of linker input statements and should be listed and interpreted for ease of handling. The CU$3REL element consists of several hundred cards, most of which are 9200/9300 object code and need not be interpreted or listed. Each deck contains sequence numbers, beginning with 0001, in columns 77 to 80.

Although each of the decks described is a ready-for-use 9200/9300 run stream, the following observations should be made concerning changes which you may wish to make in each deck.

- The CU$3REL run stream must be run before the CU$3Lxx run stream. Card 4 names SCRATCH1 on logical unit zero as its output file; this card may be changed as necessary to reflect your configuration. The output file, on 8411 discs, must consist of at least one track for directory and 40 tracks for data; on 8410, 7 sectors for directory and 7 tracks for data.

- Each of the linker run streams expects its input to be in SCRATCH1 on logical unit 0 (output from the librarian run), its output to be added to SYSFILE on zero, and its scratch file to be SCRATCH2 on logical unit 1. Card 4 of the appropriate linker run stream may be changed if the use of these files is not desired.

- Card 5 of each linker run stream is a PRGM statement which causes the UNLOAD program to be linked at the same base address as the linker used. If this is not desirable, the * may be changed to the desired link address.

- Card 7 of each linker run stream is an INCLUDE statement which causes a bar printer IOCS module to be included in UNLOAD. This card may be changed to include printer module of your choice. The SPERRY UNIVAC supplied modules were generated with the following macro calls:

```
PRINT DTFPR BKSZ = 96, PROV=PROV, CNTL=YES (for UNLOAD10)

PRINT DTFPZ BKSZ = 96, PROV=PROV, CNTL=YES (for UNLOAD & UNLD1011)
```

- Toward the end of each linker run stream is an INCLUDE statement for UNLDPNCH, which is a serial punch IOCS module. If you intend to punch on a row punch or 1004 punch, you should change the module name in this statement to UNLDRPCH or UNLDPCH4, respectively.

C.1.1.6. Executing UNLOAD

When linked, UNLOAD is operable under whatever current DNCOS supervisor the customer may be using. It adjusts itself dynamically to the amount of memory available by limiting the output blocksize. If insufficient memory exists for transcribing the current file's blocksize, appropriate diagnostics are produced and job cancellation occurs. This situation is unlikely to occur since customers who must operate in a limited amount of main storage normally use very small blocksizes. Because of its segmented structure, UNLOAD is probably smaller than most of your programs which operate upon the files to be transcribed. Figure C—4 shows a simplified diagram of UNLOAD's structure during execution. Each overlay segment is "fetched" each time it is needed, thus simplifying initialization and DTF-building logic.

A 9200/9300 system of 5.0 or higher must be used with the UNLOAD conversion utility program.

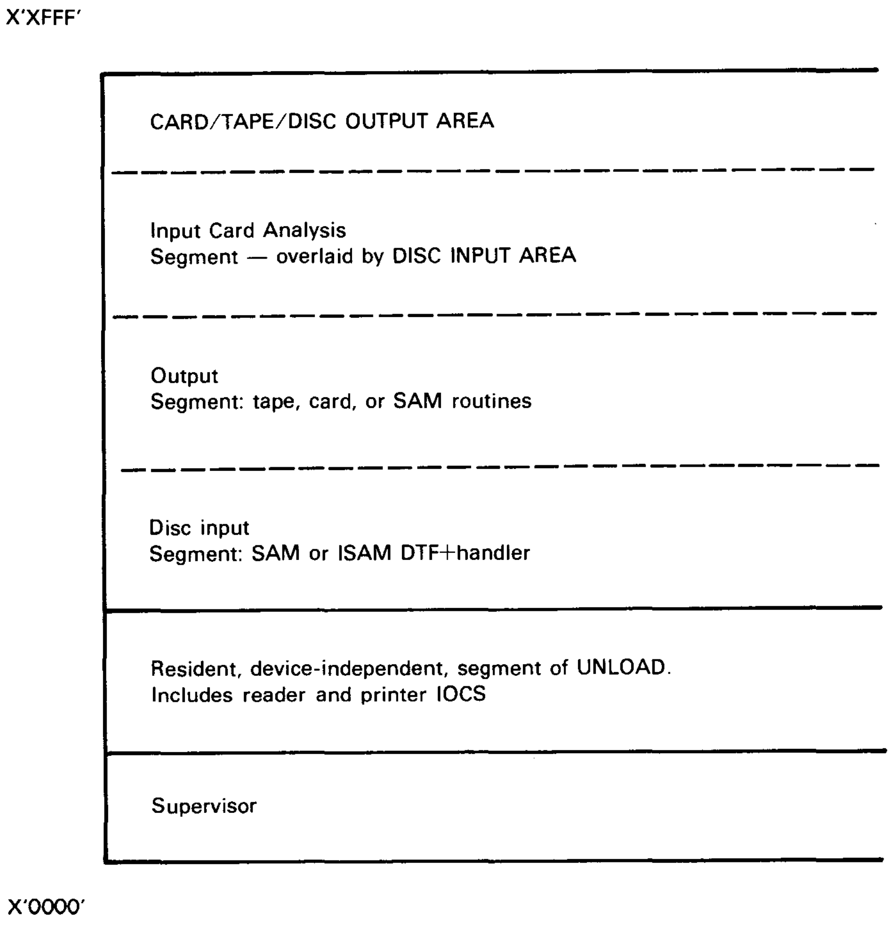

X'XFFF'
CARD/TAPE/DISC OUTPUT AREA
Input Card Analysis
Segment — overlaid by DISC INPUT AREA
Output
Segment: tape, card, or SAM routines
Disc input
Segment: SAM or ISAM DTF+handler
Resident, device-independent, segment of UNLOAD.
Includes reader and printer IOCS
Supervisor
X'0000'

Figure C—4. UNLOAD Structure During Execution

A job stream such as the one shown in Figure C—5 is prepared to convert a series of files.

```
/    JOB OS/3 DATA CONVERSION

/    DATE '070175'

/    EXEC UNLOAD

/    PARAM statements as required (See C.1.1.3)

/    DATA T

F

F    one F card per file to be converted

F

H    to cause halt for mounting new input pack

F

F    more F and H cards, ad infinitum

F

/*

/    FINIS
```

Figure C—5. Typical Job Control Stream for File Conversion

C.1.1.7. Error Processing

During the execution of UNLOAD, various halt displays may occur to indicate that an error has occurred or simply to notify you that you must take some action before further processing can take place. Before UNLOAD issues a MSG macro to stop processing, it prints the following:

!!!!!!!! MSG 77xx.

and some brief description of the error.

Note that all MSGs which are unique to UNLOAD begin with 77. This will distinguish them from standard IOCS displays which all begin with 2 or 6. Some displays inform you of an unrecoverable error situation; when the operator presses START, UNLOAD will cancel the job; these displays are starred. For all other displays you are given two and sometimes three choices. One of these choices is always to cancel the job; you can cause cancellation by replying 01. If there is only one other choice, it is indicated below and you can take this action by making any reply other than 01. If there are two choices besides cancellation, both choices are indicated; you get the first option by replying 00 (pressing START) and the second by making a reply of 02 or greater.

Table C—6 lists the halts produced by UNLOAD, the accompanying printer description, and the possible recovery actions, if any. Additional comments have been added, within parentheses, where the printer description may not be completely self-explanatory.

Table C—6. UNLOAD Error Messages (Part 1 of 2)

MSG	Explanation and Action
7700	INPUT DEVICE NOT A DISC UNIT. Correct the "F" card, clear out the reader, and refeed the run stream with the corrected card first. Press START.
7701	UNRECOGNIZED CARD TYPE. As for 7700. (Card doesn't have "H" or "F" in column 1.)
7702	INVALID FILE TYPE. As for 7700. (Column 35 didn't contain I, S, or D.)
7703	OUTPUT DEVICE TYPE INVALID. As for 7700
7704	L.U. # INVALID. As for 7700 (Could be either the input or output L.U. doesn't contain valid hex digits, or is beyond range of unit numbers available in this configuration)
7705	INSUFFICIENT MEMORY TO PROCESS THIS BLOCKSIZE. As for 7700. (If the blocksize you specified was really correct, there is no immediate recovery.)
7710	RECORD FORMAT INVALID FOR THIS TYPE OF FILE. As for 7700
7711	BLOCKSIZE NOT A MULTIPLE OF RECORDSIZE. As for 7700
7712	INPUT FILE NOT ON VTOC. Mount the right disc pack and UNLOAD retries the search (00 reply), or as for 7700 (02).
7713	READ ERROR OR INVALID VTOC RECORD ON address, DRIVE x. As for 7712.
7714	FIRST VOLUME OF INPUT FILE NOT MOUNTED. If input file is SAM, mount the correct disc pack OR accept whatever volume is mounted. If input file is ISAM, you do not have the second option.
7715	KEY LENGTH SPECIFICATION IS INVALID. As for 7700
7716	KEY LENGTH PLUS KEY LOCATION GREATER THAN RECORDSIZE. As for 7700
7717	RECORD SIZE INVALID. As for 7700 (8410 ISAM requires that record size be equal to or greater than key length and that the sum of record size and key length be no greater than 155.)
7719*	RTRV ERROR. (This error could occur following an attempt to resume reading an ISAM file following a read error.)
771A	NUMBER OF VOLUMES SPECIFIED INCORRECTLY. As for 7700

Table C—6. UNLOAD Error Messages (Part 2 of 2)

MSG	Explanation and Action
771E	INPUT ERROR $\left\{\begin{matrix} \text{cchh} \\ \text{ttss} \end{matrix}\right\}$ Close files as though end-of-file had occurred or attempt to resume reading the input file after the point where the read error occurred. (Taking the second option will result in loss of data; when reading 8411 SAM, we attempt to resume reading data on the next track. For 8410/8411 ISAM, we issue a SETL macro to attempt to retrieve a record with a key greater than the one for which the error occurred. The 8410 SAM was generated with ERRO=SKIP: after the disc dispatcher halt, the sector in error will be bypassed "automatically" by SAM.)
771F	END OF INPUT VOLUME. Remove the pack containing the current volume of the SAM file and mount the one containing the next volume OR close the file. (The latter option may be desirable for a DAM file or for a SAM file for which you do not wish to copy all volumes. This halt occurs in addition to the usual SAM EOV halts.)
7720	CAN'T READ VOL1/FORMAT-4/FORMAT-5 LABEL. Mount a different pack for output.
7721	WRONG OUTPUT PACK. VSN=xxxxxx. As for 7700. (This display can only occur when UNLOAD tries to open the first volume of an output file.)
7722*	WRITE ERROR ON VTOC. ADDR=000 xx. (This would probably indicate that the output pack is defective and should be reprepped.)
7723	OUTPUT VTOC NOT ON CYLINDER ZERO. As for 7720
7724	OUTPUT VTOC FULL. As for 7720. (You may not have allocated sufficient space on your VTOC. Remember that each OS/3 disc file requires at least two VTOC labels.)
7725	FILE ALREADY ON OUTPUT VTOC. As for 7720. (This is your protection against having two files with the same label on an output pack.)
7726	VSN xxxxxx NOW MOUNTED. Go ahead and use this pack or mount a new pack. (This message will always occur when you have mounted a new volume; it will also occur after you mount a different pack in response to various other 772x displays.)
7728	NO MORE FILE SPACE AVAILABLE ON OUTPUT PACK. As for 7720.
7729	BLOCKING FACTOR > 255. As for 7700. (This is a restriction imposed by OS/3.)
772E	OUTPUT DISC ERROR
7788	REMOVE CARDS FROM PUNCH. Continue. (This permits you to distinguish one punched output file from another and to mark output decks if desired.)
7799*	TAPE OUTPUT ERROR. (You probably are trying to use a defective tape for output.)

C.1.1.8. Hardware Requirements

The 9200/9300 UNLOAD program requires a 9200/9300 processor with sufficient main storage in which to execute. For transferring 8411 disc-to-disc, 8410 disc-to-tape, or disc-to-cards, a 16K machine suffices. For transferring 8411 disc-to-tape, or 8410 disc to 8411 disc, 24K is required. UNLOAD optionally may be emulated upon any 90/30 system that supports the 9200/9300 hardware and software configuration being emulated.

The 9200/9300 configuration must include:

- At least one 8410, 8411, or 8414 disc unit (minimum marketable 8410 or 8414 configuration is two drives)
- A card reading device supported by DNCOS, i.e., 0711, 0716, 1001, or 1004 reader
- A standard 63-character bar printer (The user may substitute his own printer IOCS module for the SPERRY UNIVAC-supplied module if he so desires.)
- One or more UNISERVO VI-C and/or UNISERVO 12 tape units if tape output is desired. If the tape units are 7 track drives, they must be equipped with the data conversion feature. All UNLOAD output to 7-track units must be in data conversion mode.

C.1.2. Disc File RELOAD Program

The RELOAD program exists as a load module in the OS/3 system load library and runs on the 90/30 system under OS/3. As mentioned earlier, RELOAD takes the intermediate files produced by the UNLOAD program and re-creates those files on a 90/30-supported disc. (The format of your existing files is not acceptable to the OS/3 data management.) File identification is specified through job control statements, and any attributes of the file you want RELOAD to produce are specified through PARAM statements. Refer to the job control user guide, UP-8065 (current version) for details of the OS/3 job control. Only one 9200/9300 file is produced for each execution of the RELOAD program.

C.1.2.1. Intermediate Files — Input to RELOAD

The intermediate file can take either of two forms: card or magnetic tape. It is produced as a result of the UNLOAD program (C.1.1) through which you have dumped your existing file to a medium that is acceptable to the RELOAD program.

C.1.2.1.1. Card File

The first two cards of the intermediate file are the two directory records. These records contain information about the input file. The remaining cards contain the actual records of the input file. A single logical record of the input file is in variable unblocked format and, in general, spans over several physical cards. RELOAD reconstructs the original record by reading as many physical records as necessary. Except for directory cards, all cards carry hexadecimal sequence numbers that RELOAD checks for consecutive sequencing. (You can inhibit this function of RELOAD by use of the SEQCHK keyword parameter in your PARAM statements.) An incorrect sequence number creates an error condition. The offending card is processed nevertheless, but a sufficient number of such errors will result in the termination of the conversion function.

C.1.2.1.2. Tape File

The general form for the intermediate file is a multifile, multitape file. The first block of the file contains the two directory records, and the remainder of this file contains the records of your input file. Regardless of the record format of the input file, the intermediate file records are always in variable, blocked format. However, this format is transparent to you. In addition, the intermediate file is rewound when it is OPENed or CLOSEd. You do not have to position the tape to the beginning of the file. This is actually a responsibility of OS/3 data management which is invoked as a function of the RELOAD program.

C.1.2.2. Conversion Mechanism for RELOAD

The RELOAD program uses OS/3 data management to produce files and support the following access methods: sequential (SAM), direct (DAM), and indexed sequential (ISAM). These methods are discussed in detail in the OS/3 data management user guide, UP-8068 (current version). Communication between RELOAD and UNLOAD is established by two directory records written at the beginning of the intermediate file. These records contain the attributes of the input file and are followed by the records of the input file itself. If you wish, you may make changes to the attribute values specified or add and delete attributes by use of PARAM statements. The following list summarizes the nature of the output file attributes:

- File type
  - SAM
  - DAM
  - ISAM

- Record type
  - Fixed unblocked
  - Fixed blocked
  - Variable unblocked
  - Variable blocked
- Block size
- Record size (fixed-records only)
- Key size (ISAM only)
- Key location (ISAM only)
- Percentage overflow (ISAM only)

In general, it is possible to convert any set of attributes and their related values to any other set of attributes with new or unchanged values. However, certain combinations of conversion are prohibited. These restrictions are discussed later in this paragraph under the explanation of parameters. Attributes that you do not wish changed need not be specified at all in the PARAM cards you submit to RELOAD. However, before executing RELOAD, you are urged to refer to the OS/3 data management user guide, UP-8068 (current version) for record formats of SAM, DAM, and ISAM files. The OS/3 record formats are, in general, different from those of your existing system. This also is true of the terminology used in describing OS/3 records. Block size is a good example of this. In OS/3, block size always refers to the size of the block, not including the 8-byte count field. Depending upon the type of conversion, it may be necessary for you to specify certain file attributes on the PARAM cards.

C.1.2.3. PARAM Card Preparation

The PARAM cards are submitted at RELOAD time and are used to inform RELOAD of the attributes for the output file being produced. Preparation of these cards is your responsibility. All of the parameters have default values so that you need only specify those parameters that differ from the default.

In making up the PARAM card, the following rules apply:

- The keyword parameters may be coded in any order. Columns 72 through 80 are ignored.
- Any number of parameters may be specified on a PARAM card, but parameters may not be broken over card boundaries.

- If keywords are duplicated, the last one read is considered valid.
- The characters PARAM must be preceded and followed by at least one blank (△).
- Keywords are separated by commas with no embedded blanks.
- The first blank detected in the operand field terminates the scan for that card.
- Numeric values are expressed in decimal.
- Capital letters, commas, equal signs, and parentheses must be coded exactly as shown.
- Lowercase letters and words are generic terms representing information that must be supplied by the user.
- Information contained within braces represents mandatory entries of which one must be chosen.
- Information contained within brackets represents optional entries that (depending upon program requirements) are included or omitted. Braces within brackets signify that one of the specified entries must be chosen if that parameter is to be included.

Format:

LABEL	△OPERATION△	OPERAND
//	PARAM	[BLKL=n]
		$\left[\text{ERRCOUNT=}\left\{\begin{matrix}\text{0–32767}\\ \text{0}\end{matrix}\right\}\right]$
		$\left[\text{FTYP=}\left\{\begin{matrix}\text{D}\\ \text{I}\\ \text{S}\end{matrix}\right\}\right]$
		$\left[\text{IFIL=}\left\{\begin{matrix}\text{C}\\ \text{T}\end{matrix}\right\}\right]$
		[KLOC=n]
		[KSIZ=3–253]
		$\left[\text{OVFL=}\left\{\begin{matrix}\text{0–80}\\ \text{20}\end{matrix}\right\}\right]$
		[RECL=n]
		$\left[\text{RTYP=}\left\{\begin{matrix}\text{FB}\\ \text{FU}\\ \text{VB}\\ \text{VU}\end{matrix}\right\}\right]$
		$\left[\text{SEQCHK=}\left\{\begin{matrix}\text{NO}\\ \text{YES}\end{matrix}\right\}\right]$

BLKL Keyword Parameter:

BLKL=n

A decimal number specifying the block length of the output file. It does not, however, include the 8-byte count field. It allows you to have an output block size that is different from the input block size. If you are using the fixed-sector 8416 disc, you must specify a block size that is a multiple of 256. For fixed records in the output file, the block and record size (whether specified by PARAM card or assigned by default) must maintain the following relationship:

1. $b = r$ (DAM)
2. $b = nr$ (SAM)
3. $b = n(r+5)+2$ (ISAM)

where b is block size, r is record size, and n is the blocking factor (see OS/3 data management user guide, UP-8068 (current version) for details).

If omitted, the output block size is assumed to be the same as the input block size unless the following conditions exist:

a. Input file is an ISAM file with fixed records.

b. Output file is an ISAM file with fixed records (whether coded on PARAM cards or assigned by default).

c. Both BLKL and RECL parameters are not specified.

In these cases, RELOAD assigns default values determined by the following:

a. Output record size is assumed equal to the input record size. (See RECL parameter.)

b. Input blocking factor is used to compute the default output block size such that the special OS/3 block size and record size relationship holds.

This facility allows you to convert your ISAM file without supplying any PARAM cards to RELOAD.

ERRCOUNT Keyword Parameter:

ERRCOUNT= $\left\{ \begin{matrix} 0\text{–}32767 \\ 0 \end{matrix} \right\}$

For a tape intermediate file, the value specified determines the number of nonfatal errors that can be tolerated during file processing. When a nonfatal error occurs during tape read, the offending block is skipped, and a new block is read (all records in your input file, therefore, may not appear in your output file).

For a card intermediate file, the value specified determines the number of sequence check errors that can be ignored during card processing. The contents of out-of-sequence cards are processed, however. Refer to C.1.2.6.1 for a discussion of nonfatal errors. All errors encountered during directory read are considered fatal and result in the termination of file conversion.

If omitted, zero is assumed, and the first nonfatal error encountered causes file processing to be suspended and the file closed.

FTYP Keyword Parameter:

FTYP= $\left\{\begin{matrix}\mathbf{D}\\\mathbf{I}\\\mathbf{S}\end{matrix}\right\}$

Alphabetic characters specifying the type of output file being created.

D

Creates a direct access (DAM) output file.

I

Creates an index sequential (ISAM) output file.

S

Creates a sequential (SAM) output file.

If omitted, the output file type is the same as the input file.

IFIL Keyword Parameter:

IFIL= $\left\{\begin{matrix}\mathbf{C}\\\mathbf{T}\end{matrix}\right\}$

An alphabetic character defining the medium that contains the intermediate file.

C

Represents card file.

T

Represents tape file.

If omitted, a tape intermediate file is assumed.

KLOC Keyword Parameter:

KLOC=n

A decimal number specifying the key location for an ISAM output file. The value specified represents the position of the key field from the beginning of the record, starting from zero. It includes the two bytes of the record descriptor for variable-length records.

If omitted, the value for this parameter is obtained from the input file attributes. If the input file is not an ISAM file, a value of zero is assumed for fixed-length records and a value of 2 for variable-length records.

KSIZ Keyword Parameter:

KSIZ=3–253

Specifies the key size for an ISAM output file.

If omitted, the key size is assumed to be the same as the input file, provided that the input file is an ISAM file. If the key size of the input file is unobtainable, RELOAD terminates.

OVFL Keyword Parameter:

OVFL= $\left\{\begin{matrix}\mathbf{0\text{–}80}\\ \mathbf{20}\end{matrix}\right\}$

Specifies the percentage of disc space to be used as an overflow area for an ISAM output file.

RECL Keyword Parameter:

RECL=n

A decimal number specifying the output record length for fixed-length records. If the output record size is larger than input records (variable or undefined length to fixed-length conversion) or larger than all input records (variable, undefined, or fixed length to fixed-length conversion). RELOAD pads the output records with binary 0's. If the output record size is smaller than any input record, RELOAD terminates as soon as such a condition is encountered.

If omitted, the record length is assumed to be the same as the input record length provided that the input file is a fixed length file.

For DAM output files, the record size is assumed to be the same as the input record size, provided that the input records are fixed length. If the input file has variable-length records, the output record length is unavailable, and RELOAD terminates.

RTYP Keyword Parameter:

RTYP= $\left\{\begin{matrix}\mathbf{FB}\\ \mathbf{RU}\\ \mathbf{VB}\\ \mathbf{VU}\end{matrix}\right\}$

Defines the output file record type. This keyword must be specified if the input file record type is undefined.

FB

Specifies fixed, blocked records.

FU

Specifies fixed, unblocked records.

VB

Specifies variable, blocked records.

VU

Specifies variable, unblocked records.

Parameters FB and VB may not be specified for a DAM output file, and FU and VU may not be specified for an ISAM output file.

If omitted, the output file record type is assumed to be the same as the input file record type.

SEQCHK Keyword Parameter:

SEQCHK= $\left\{\begin{array}{l}\textbf{NO}\\\textbf{YES}\end{array}\right\}$

Specifies that you do not want RELOAD to check the card sequence number of a card intermediate file.

SEQCHK=NO

Specifies that you do not want RELOAD to check the card sequence numbers of the card intermediate file. Out-of-sequence cards will not, in this case, initiate the processing of nonfatal errors.

SEQCHK=YES

Specifies that you want RELOAD to check sequence numbers for a card intermediate file. A message is printed for each out-of-sequence card.

If omitted, RELOAD automatically performs card sequence checking and warns you as soon as an out-of-sequence card is detected. RELOAD adjusts itself to the new sequence and processes the out-of-sequence card. The number of nonfatal errors permitted or ignored is determined by whatever you have specified for the ERRCOUNT parameter.

C.1.2.4. Job Control Interface to RELOAD

All files used by RELOAD must be assigned through job control. This includes such files as printer, tape, and disc file. Job control statements are not needed for a card intermediate file since it is read by GETCS. Job control also reserves disc space.

There are five job control statements that can be used for device assignments. See the OS/3 job control user guide, UP-8065 (current version) for a complete description of each statement. The following is a brief discussion of statements and conventions as applicable to RELOAD.

■ DVC statement

This statement is used to request the assignment of a peripheral device to a job. It is required for printer, tape, and disc files. Only one DVC statement is used for a SAM tape or disc single volume or multivolume file. For a DAM or ISAM disc file, a DVC statement must be used for each volume of the file.

- VOL statement

  This statement supplies the volume serial numbers of the volumes to be accessed by the job. It is required for tape and disc files. For a SAM tape file, all volumes used by the file must be specified on a single VOL statement. For a DAM/ISAM disc file, each volume must be specified separately on VOL statements.

  NOTE:

  All volumes for a DAM or ISAM disc file must be online. For a SAM tape or disc file, only one volume must be online.

- EXT statement

  This statement is used for initial allocation of space for a disc file. Disc space is assigned as soon as this statement is encountered by job control. Subsequent runs referring to the same file may not have this statement. This is true even if a rerun of RELOAD is required.

- LBL statement

  This statement is used to supply label information for files on disc and tape volumes. For an output disc file produced by RELOAD, a file identifier of up to 44 characters must be specified on the LBL statement. This may or may not be the same as the file-id of your original disc file.

  For the intermediate tape file, the first 17 characters of the 44-character file-id of your original input file must be used as file-id. In the case of multivolume, multifile tape, a file sequence number also must be specified on the LBL statement. This is used to position the tape to the correct file and is done automatically by data management. The file sequence number specifies the position of this file with respect to the first file within a multifile set. This number can be obtained from the UNLOAD printout.

- LFD statement

  This statement is used to link the file information in the control stream to the data management file defintiions. For the RELOAD output (disc) file, the filename CU$LOAD must be used, and positional parameter 3 on the LFD statement must specify the option INIT. This option indicates that the specified file is to be initialized and is required if RELOAD is rerun for the same file. Once the conversion is complete, however, you should omit the INIT parameter for all subsequent use of the file. The file name to be stated on the LFD statement for an intermediate tape file is RE$LOAD, and the file name for a printer file is PRNTR.

Table C—7 summarizes the job control statements required by the files associated with RELOAD.

Table C—7. Job Control Statements Associated With RELOAD

Job Control Statement	File Type			
	Intermediate Tape	Intermediate Card	Output Disc	Output Printer
DVC	X		X	X
EXT			X	
LBL	X		X	
LFD	X		X	X
VOL	X		X	

C.1.2.5. Job Control Stream Examples for RELOAD

The following examples show file conversion for three hypothetical situations using the RELOAD program.

- Example 1: ISAM-to-ISAM File Conversion

  In this example, you want to convert an ISAM input file to an OS/3 ISAM output file. Assume that your input file has the following attributes:

  — File type = ISAM

  — Record type = Fixed blocked

  — Block size = 200

  — Record size = 50

  — Blocking factor = 4

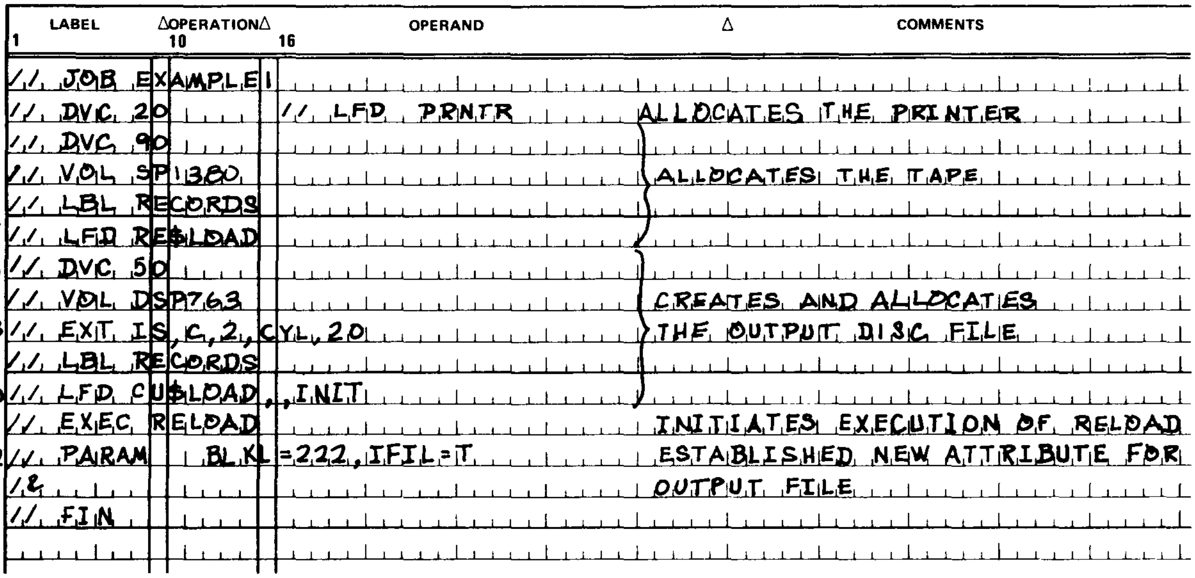
  — Keysize = 20

  — Key location = 5

  Also assume that you have used the 9200/9300 UNLOAD program to create a single-volume intermediate tape file that is used as the input to RELOAD and you want to create an OS/3 file that has the same attributes as your original file. In order to maintain the same blocking factor, a block size of 222 must be specified for the output file. This is computed by the formula:

  Block size = blocking factor (record size + 5)+2
  = 4 (50+5)+2
  = 222

The job stream required to perform this conversion is as follows:



```
    LABEL     ΔOPERATIONΔ           OPERAND                   Δ              COMMENTS
1              10     16
   // JOB EXAMPLE1
 1 // DVC 20             // LFD  PRNTR             ALLOCATES THE PRINTER
 2 // DVC 90                                      }
 3 // VOL SP1380                                  }ALLOCATES THE TAPE
 4 // LBL RECORDS                                 }
 5 // LFD RE$LOAD                                 }
 6 // DVC 50                                      }
 7 // VOL DSP763                                  }CREATES AND ALLOCATES
 8 // EXT IS,C,2,CYL,20                           }THE OUTPUT DISC FILE
 9 // LBL RECORDS                                 }
10 // LFD CU$LOAD,,INIT                           }
11 // EXEC RELOAD                                  INITIATES EXECUTION OF RELOAD
12 // PARAM   BLKL=222,IFIL=T                      ESTABLISHED NEW ATTRIBUTE FOR
   /&                                              OUTPUT FILE
   // FIN
```

1. Allocates the printer

2—5. Allocates the tape by specifying:

   2. Device number
   3. Volume serial number
   4. Input (original) file-id (first 17 characters)
   5. File name required by RELOAD for intermediate file

6—10. Creates and allocates the output disc file by specifying:

   6. Output disc device number
   7. Output disc volume serial number
   8. Disc area assignment for new file
   9. Input (original) file-id that is now assigned to the new output file
   10. File name required by RELOAD for the output file.

11. Initiates execution of RELOAD program

12. Informs RELOAD that the output file must have a block size of 222 and that the intermediate file is on tape. The remaining attributes are assumed by default. Nonfatal errors are not tolerated. This PARAM card could have been omitted without any change in the results.

- Example 2. SAM-to-DAM File Conversion

In this example, you want to convert a large SAM input file to a DAM output file. The input and output files have the following attributes:

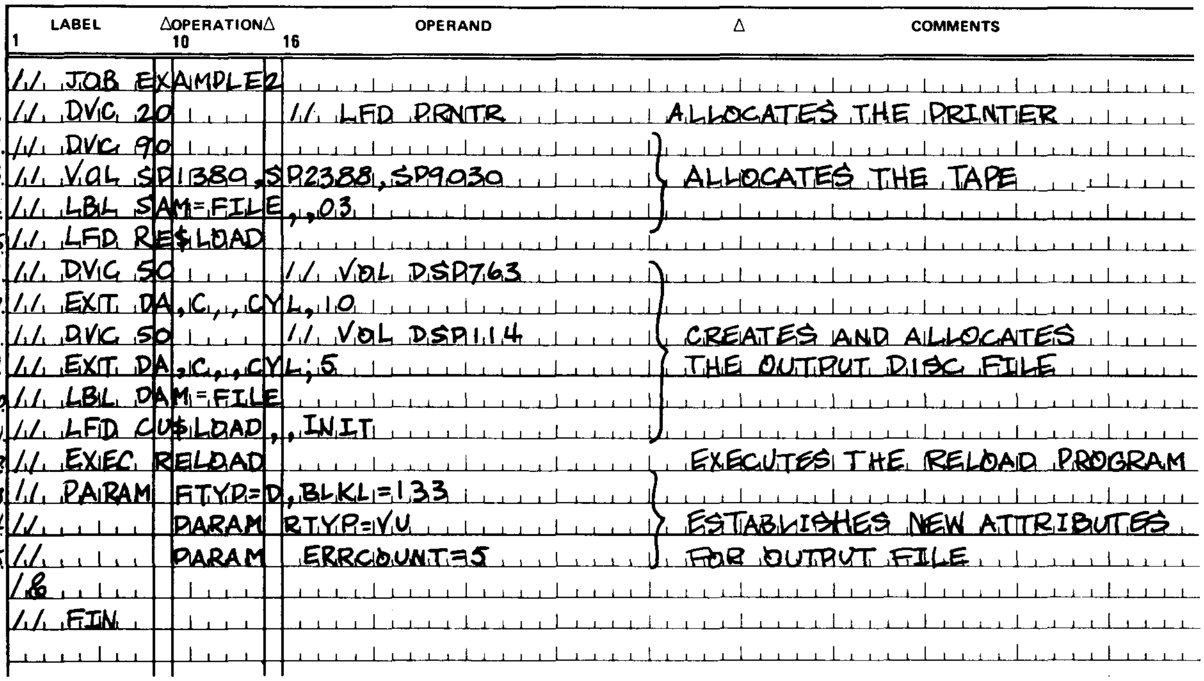
Attributes	Input file	Output file
File type	SAM	DAM
Record type	Fixed blocked	Variable unblocked
Block size	500	133
Record size	125	—

Assume that the intermediate file was created by 9200/9300 UNLOAD as a multivolume, mutifile tape with a file sequence number of 3. The output is to be written to a multivolume disc file, and at most five nonfatal errors can be tolerated. The computation for block size is:

$$\begin{aligned}\text{block size} &= 125+4+4\\ &= 133\end{aligned}$$

Note that block size includes four bytes of the record descriptor word (RDW) and four bytes of the block descriptor word (BDW).

The job stream required to perform this conversion is as follows:



```
 LABEL      ΔOPERATIONΔ        OPERAND                          Δ            COMMENTS
1           10         16
// JOB EXAMPLE2
// DVC 20          // LFD PRNTR                      ALLOCATES THE PRINTER
// DVC 90                                           }
// VOL SP1380,SP2388,SP9030                         } ALLOCATES THE TAPE
// LBL SAM=FILE,,03                                 }
// LFD RE$LOAD
// DVC 50          // VOL DSP763                    }
// EXIT DA,C,,CYL,10                                }
// DVC 50          // VOL DSP114                    } CREATES AND ALLOCATES
// EXIT DA,C,,CYL;5                                 } THE OUTPUT DISC FILE
// LBL DAM=FILE                                     }
// LFD CU$LOAD,,INIT                                }
// EXEC RELOAD                                        EXECUTES THE RELOAD PROGRAM
// PARAM FTYP=D,BLKL=133                            }
//         PARAM RTYP=VU                            } ESTABLISHES NEW ATTRIBUTES
//         PARAM  ERRCOUNT=5                        } FOR OUTPUT FILE
/&
// FIN
```

1. Allocates the printer

2—5. Allocates the tape by specifying:

2. Device number
3. Volume serial numbers for the tapes
4. File-id and file sequence number
5. File name for the intermediate file as required by RELOAD

6—11. Creates and allocates a multivolume DAM disc file by specifying:

6 and 8. Output disc device numbers
7. A 10-cylinder DAM extent for DSP 763
9. A 5-cylinder DAM extent for DSP 114
10. The file-id for the output file (which is different from the input file by choice)
11. The file name for the output file as required by RELOAD

12. Initiates the execution of RELOAD program

13. Informs a RELOAD to create a DAM output file with a block size of 133

14. Requests a variable unblocked output file

15. Establishes a maximum of five nonfatal errors to be tolerated before RELOAD terminates.

- Example 3. ISAM-to-SAM File Conversion

In this example, you want to convert an ISAM input file to an output SAM file that has a large block size. The input and output files have the following attributes:

Attributes	Input file	Output file
File type	ISAM	SAM
Record type	Fixed unblocked	Fixed blocked
Block size	1000	6000
Record size	40	50
Key size	15	—
Key location	0	—

Assume that the intermediate file is on cards and a multivolume output file is to be created. Since the record size is being increased from 40 to 50, RELOAD will pad each logical record (on the right) with 10 bytes of binary 0's. The job stream required to perform this conversion is as follows:

```
    LABEL     ΔOPERATIONΔ             OPERAND                              Δ
1              10     16
1.  // JOB EXAMPLE3,,6000
2.  // DVC 20 // LFD PRNTR
3.  // DVC 50
4.  // VOL DSP763
5.  // EXT SQ,C,,CYL,5
6.  // VOL DSP114
7.  // EXT SR,C,,CYL,10
8.  // LBL LARGE*BLOCK*FILE
9.  // LFD CU$LOAD,,INIT
10. // EXEC RELOAD
11. // PARAM IFIL=C,SEQCHK=YES,FTYP=S,BLKL=6000,RECL=50
12. /&
13. // FIN

14.          UNLOAD CARD OUTPUT

15. /*
```

1. Requests extra space to enable RELOAD to handle larger output block size

2. Assigns the printer

3—9. Allocates the discs and assigns space for the output file by specifying:

   3. Device number
   4. Volume serial number of the first volume of a SAM file
   5. A five cylinder extent defined for the first volume
   6. Volume serial number of the second volume of a SAM file
   7. A 10-cylinder extent defined for the second volume
   8. Input file-id assigned to the output file
   9. File name for the output file as required by RELOAD

10. Initiates execution of RELOAD

11. Inform RELOAD that the intermediate file is on cards, sequence checking is to be performed, output file type is SAM, the block size is 6000, and record size is 50

12. End of job

13. Ending card reader operation

14. Card output from RELOAD

15. Indicates end of data

C.1.2.6. Error Processing Performed by RELOAD

The RELOAD program provides you with certain error processing options during the conversion of your files. The options available to you are dependent upon the file type. The number of errors that may be ignored before the conversion process is terminated by RELOAD is established by you in the form of PARAM card specifications. When file processing is terminated upon occurrence of an error condition, all files are CLOSEd and the partial file generated at this point may be usable.

C.1.2.6.1. Error Processing for Intermediate Files

Two types of errors are recognized during the processing of intermediate files: fatal and nonfatal errors. Fatal errors are those which terminate further file processing. Nonfatal errors allow file processing to continue by either ignoring the error condition or by skipping the offending block. You can establish the maximum number of nonfatal errors that can be tolerated through the ERRCOUNT keyword parameter in your PARAM card to RELOAD. To make this function work, however, you must reply with a keyin of the character U to any read/write error message that appears on the system console. If you cannot afford a bad or incomplete file, it behooves you to specify an ERRCOUNT=0. All error conditions, fatal and nonfatal, cause appropriate error messages to be printed.

- Card Intermediate File

  A card sequence error is the only nonfatal error for a card intermediate file. If you have chosen to inhibit the sequence check function (see SEQCHK parameter), RELOAD will not perform sequence checking, and you will not be informed of any sequence errors. In either case, the card with the incorrect sequence is processed and contributes to the construction of the original logical record. Two conditions could arise as a result of incorrect card sequence. First, the conversion function is terminated if you have chosen not to tolerate any nonfatal errors by not specifying the ERRCOUNT keyword parameter at all or by specifying ERRCOUNT=0. The conversion function is also terminated if the number of card sequence errors exceeds the number specified by the ERRCOUNT parameter. Second, it is possible, if the first condition does not occur, that a subsequent logical record may not be constructed as a result of a card sequence error. This gives rise to a record-invalid condition (fatal error).

  Upon encountering an incorrect sequence, RELOAD adjusts itself to the new sequence. It is important to mention that an out-of-sequence card may result in the construction of a bad logical record. Directory records do not have sequence numbers.

Remember, fatal errors terminate all file processing, and the following conditions are considered fatal.

- Hardware error
- Premature end of file. This error occurs when a partial UNLOAD punched card is supplied to RELOAD (including PARAM cards).
- Invalid record. This occurs if a logical record cannot be formed due to an incomplete RELOAD deck or an invalid record length.

■ Tape Intermediate File

RELOAD considers the following list of errors as nonfatal errors.

- Hardware errors
- Unique unit errors
- Unrecoverable errors
- Wrong length found
- Parity check
- Data converter check (7-track only)

All other errors returned by the tape IOCS are considered fatal, including errors that occur during the reading of directory records. (See data management user guide, UP-8068 (current version) for a complete list of errors.) Unless the count of nonfatal errors exceeds the ERRCOUNT specification, the block causing the error is skipped, and RELOAD attempts to read the next block.

C.1.2.6.2. Error Processing for Output Files

Records written to the output files are checked for parity. All errors in the writing of the output file, including parity errors, are considered fatal. The conversion function is terminated, and all files are CLOSEd. The partially constructed output file may be usable.

C.1.2.7. Main Storage Requirements

RELOAD reserves enough main storage to handle block sizes of up to 2K (2048) bytes. This amounts to less than 20K (20,480 decimal or X'5000') bytes of main storage and thereby permits RELOAD to run in the 32K minimum system. Additional main storage can be made available to RELOAD by the use of 'min' and 'max' memory size parameters on the JOB card (see OS/3 job control user guide, UP-8065 (current version)). If, in a particular run, RELOAD determines that the main storage available to it is insufficient, it prints out the amount of main storage required. This size can then be used on the JOB card in the next RELOAD run.

The exact main storage requirements for RELOAD depend upon the precise input and output file attributes, such as access methods (file types), record types, block sizes, record sizes (for fixed length records), and the type of intermediate file. However, an upper bound can be computed by the following formula.

$$M = 14K + BI + 2 * BO \text{ bytes}$$

where:

M
: Is the upper bound on main storage required (worse case).

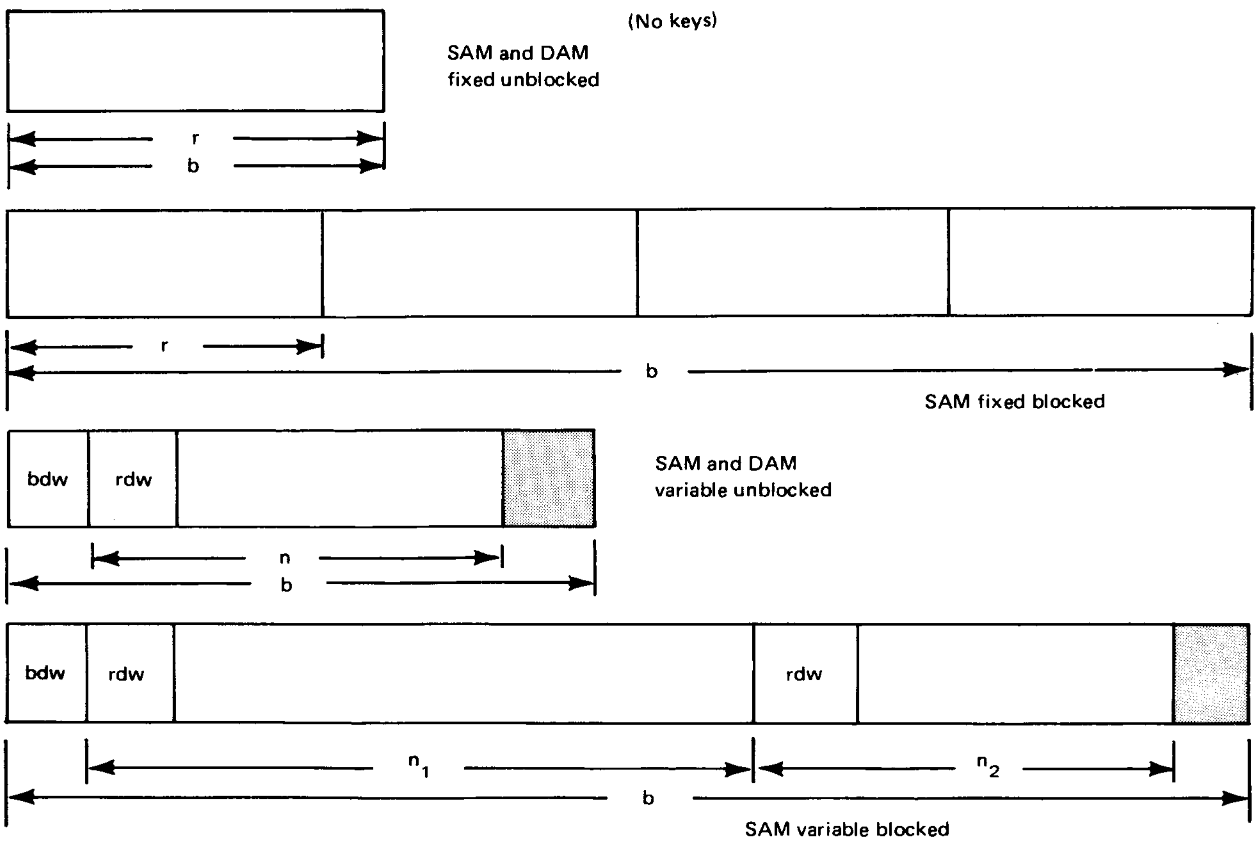
BI
: Is the input block size.

BO
: Is the output block size.

C.1.2.8. Record Format for OS/3

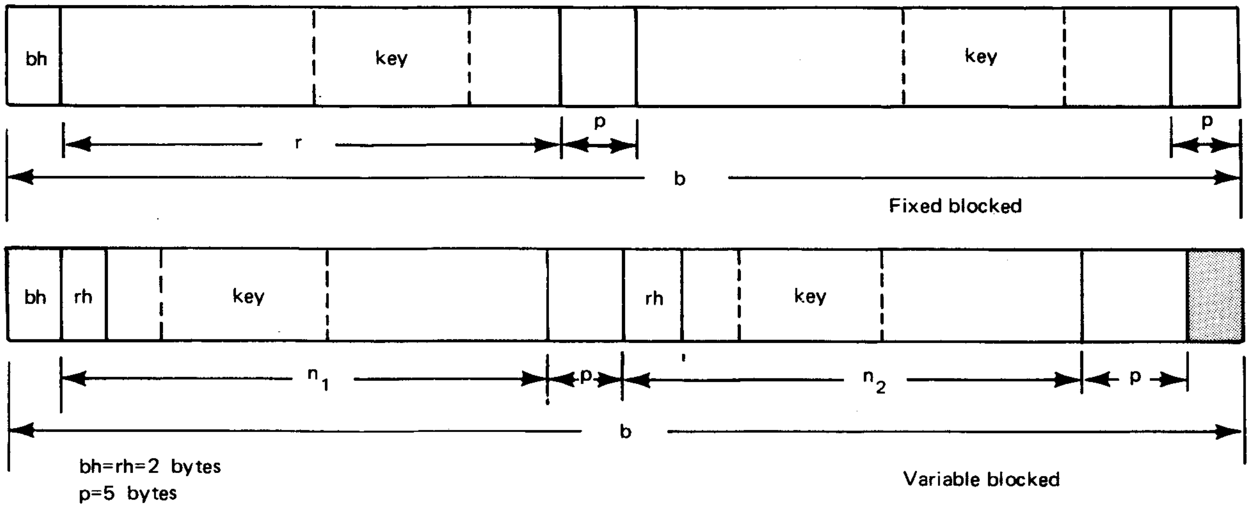
The following record formats are provided as a quick reference only. Detailed descriptions of each record format are provided in the OS/3 data management user guide, UP-8068 (current version).

NONINDEXED RECORD FORMATS



bdw=rdw=4 bytes

INDEXED SEQUENTIAL RECORD FORMATS



where:

bdw	is block descriptor word.
rdw	is record descriptor word.
bh	is block header.
rh	is record header.
p	is record pointer.
b	is block size.
r	is record size.

C.1.2.9. RELOAD Messages

The RELOAD program does not generate any console messages. However, there may be messages related to the conversion procedure that are generated by data management and space management. Some of these messages may be routine operator communications, such as mount requests, and others may be error messages. RELOAD does, however, print these types of messages: warnings, errors (fatal and nonfatal), and informational.

- Warning Messages

  These messages indicate the presence of low-severity errors. RELOAD ignores such a condition or assumes a default. Processing is allowed to continue. RELOAD warning messages are listed and described in Table C—8.

Table C—8. RELOAD Warning Messages

Message	Cause/Explanation
IFIL PARAMETER NOT SPECIFIED. TAPE INTERMEDIATE FILE ASSUMED.	RELOAD will not check for the presence of a card intermediate file.
RECORD SIZE SPECIFIED FOR VARIABLE RECORDS. CONDITION IGNORED.	Parameter RECL is not required for a variable-length output file.
OS/3 ISAM DOES NOT SUPPORT UNBLOCKED RECORDS. BLOCKED ASSUMED.	Specify RTYP=VB or FB for an ISAM output file.
PARAMETER OVFL NOT SPECIFIED. 20 ASSUMED.	OS/3 ISAM requires you to specify a value for OVFL (percentage of disc space to be used for overflow area). If not specified, 20% is assumed.
PARAMETER KLOC NOT SPECIFIED. ASSUMED 0 FOR FIXED BLOCKED, 2 FOR VARIABLE BLOCKED.	Specify KLOC parameter (key location) for ISAM output file.
OVFL PARAMETER SPECIFIED FOR SAM/DAM FILE. IGNORED.	OVFL parameter applies only to ISAM files.
KEYS NOT SUPPORTED FOR NON-INDEXED FILE. CONDITION IGNORED.	RELOAD does not support keyed SAM or DAM files.

- Error Messages

Error messages indicate the existence of fatal or nonfatal error conditions resulting from conflicting parameter specifications or conditions that are encountered during file processing. Fatal errors result in immediate termination of the file conversion process, while nonfatal errors allow the process to continue until such errors exceed the error limit you have set in the ERRCOUNT specification to RELOAD program.

The error messages associated with the RELOAD program are listed in Table C—9.

Table C—9. RELOAD Error Messages (Part 1 of 4)

Error Message	Cause/Explanation
FATAL OPEN/CLOSE READ ERROR ENCOUNTERED ON INTERMEDIATE FILE.	An OPEN/CLOSE read error has occurred that does not permit further processing. See console sheet for details.
PREMATURE END OF FILE ENCOUNTERED ON INTERMEDIATE FILE.	A null file or a file with too few records supplied to RELOAD
INVALID INTERMEDIATE FILE.	RELOAD has been supplied with a file not produced by UNLOAD.

Table C—9. RELOAD Error Messages (Part 2 of 4)

Error Message	Cause/Explanation
ERROR — PARAMETER OR ARGUMENT INVALID OR MISSPELT	The card preceding this message is not a PARAM card, has a syntax error in the statement, or has a misspelled keyword parameter. Correct the problem and rerun.
KSIZ PARAMETER SPECIFICATION ERROR. LIMITS 3—253.	Correct the argument of KSIZ parameter.
OVFL PARAMETER SPECIFICATION ERROR. LIMITS 0—80.	Correct the argument of OVFL parameter.
FOR AN INPUT FILE WITH UNDEFINED RECORDS, RTYP PARAMETER MUST BE SPECIFIED FOR THE OUTPUT FILE.	OS/3 does not support undefined length records. Specify RTYP parameter.
OUTPUT RECORD SIZE NOT SPECIFIED FOR FIXED RECORDS.	Specify RECL parameter or specify variable records.
NUMBER TOO LARGE.	A number on the PARAM card exceeds 10 digits.
MEMORY AVAILABLE INSUFFICIENT. ALLOCATE AT LEAST XXXXX BYTES (DECIMAL) TO THIS JOB.	RELOAD has determined that the conversion will require more main storage than has been allocated. Check your PARAM cards and rerun with a request for additional main storage on the JOB card.
OUTPUT BLOCK SIZE MUST EXCEED 9 FOR ISAM VARIABLE BLOCKED RECORDS.	Check BLKL parameter (or default value) and correct as necessary.
PARAMETER KSIZ MUST BE SPECIFIED FOR AN ISAM OUTPUT FILE.	RELOAD was unable to pick up a default value for KSIZ parameter. Specify KSIZ parameter and rerun.
ISAM OUTPUT RECORD SIZE MUST EXCEED KLOC + KSIZ.	Check RECL, KLOC, and KSIZ parameters (or their default values) and correct as necessary.
OUTPUT ('BLK SIZE'-2) IS NOT A MULTIPLE OF (REC SIZE+5).	This is a requirement for OS/3 ISAM fixed-length records. Check BLKL and RECL parameters (or their default) and correct as necessary.
OUTPUT BLK SIZE MUST EXCEED KSIZ + KLOC + 7 FOR VARIABLE BLOCKED RECORDS.	Check BLKL, KSIZ, and KLOC parameters (or their defaults) for the ISAM file and correct as necessary.

Table C—9. RELOAD Error Messages (Part 3 of 4)

Error Message	Cause/Explanation
KLOC FOR VARIABLE RECORDS MUST BE AT LEAST 2.	ISAM variable-length records include two bytes of record header, and this value must be reflected in key location. Check KLOC parameter (or its default) and correct as necessary.
OUTPUT BLOCK SIZE MUST EQUAL OR EXCEED XXXXX BYTES.	For a fixed-length ISAM file, block size is less than record size +7. Check BLKL and RECL parameters (or their defaults) and correct as necessary.
BLOCKED RECORDS NOT SUPPORTED BY DAM.	This is an OS/3 restriction. Specify RTYP=FU or RTYP=VU.
BLOCK SIZE DOES NOT EQUAL RECORD SIZE FOR FIXED UNBLOCKED RECORDS.	Check BLKL and RECL parameters (or their defaults) for SAM/DAM fixed unblocked files and correct as necessary.
OUTPUT BLOCK SIZE MUST BE A MULTIPLE OF RECORD SIZE FOR FIXED BLOCKED RECORDS.	Check BLKL and RECL parameters (or their defaults) for a SAM fixed-blocked file and correct as necessary.
OUTPUT BLK SIZE FOR VARIABLE RECORDS MUST EXCEED 8 BYTES.	Check BLKL parameter (or its default) for SAM/DAM variable-length records and correct as necessary.
CARD SEQUENCE ERROR.	(N)* Sequence error was detected in the card intermediate file. Processing continues if error limit is not exceeded. The contents of the offending card are processed. If you cannot tolerate any errors, rerun UNLOAD and RELOAD programs.
TAPE READ ERROR.	(N)* An error was detected in the reading of the tape intermediate file. If the error limit is not exceeded, the bad block is skipped over and the next block is read. The operator must reply with a U to a read error console message. Check the condition of tape and the tape drive. If necessary, rerun UNLOAD and RELOAD programs.
ERROR COUNT EXCEEDED.	Too many intermediate file sequence check/read errors encountered. Rerun UNLOAD and RELOAD programs. Check card file for mispunches/off-punches, or check the condition of tapes and tape drives.

*(N) indicates a nonfatal error.

Table C—9. RELOAD Error Messages (Part 4 of 4)

Error Message	Cause/Explanation
INPUT RECORD TOO LARGE. INCREASE OUTPUT REC/BLK SIZE(S).	1. For a fixed-to-fixed-length conversion, the output record size is less than the input record size. Specify a larger record size and change block size to maintain the same blocking factor. 2. For a variable or undefined-to-fixed-length conversion, the input record has a length (minus RDW length) larger than the output record size. Same action as 1. 3. For a variable-length conversion, the input record is larger than the output block size. Specify a larger block size.
DISC OPEN/CLOSE/WRITE ERROR.	For details, see console sheet and Appendix B of data management user guide, UP-8068 (current version). Correct as necessary.
INVALID RECORD	RELOAD program was unable to construct a valid logical record from card intermediate file.
INPUT BLOCK SIZE NOT SPECIFIED	RELOAD has been supplied with a file that was not produced by UNLOAD or the intermediate file directory does not contain the input block size. Rerun RELOAD.

- Information Messages

Information messages summarize file characteristics and results of the RELOAD run. The messages are both informational and statistical in nature. These messages are listed in Table C—10.

Table C—10. RELOAD Information Messages

INPUT FILE CHARACTERISTICS. (ALL NUMERIC VALUES ARE IN DECIMAL.)
OUTPUT FILE CHARACTERISTICS. (ALL NUMERIC VALUES ARE IN DECIMAL.)
FILE NAME
FILE TYPE
BLOCK LENGTH
RECORD LENGTH
KEY LENGTH
KEY LOCATION
PERCENTAGE OVERFLOW
CREATION DATE
EXPIRATION DATE
FILE LOAD STARTED
NUMBER OF LOGICAL RECORDS READ
NUMBER OF LOGICAL RECORDS WRITTEN
NUMBER OF NONFATAL ERRORS
UTILITY ABORTED. FILE NOT LOADED
FILE LOAD SUCCESSFUL

C.2. BASIC ASSEMBLY LANGUAGE TRANSLATION BY SOURCE CODE ANALYZER ROUTINE

The source code analyzer (SCAN) is a basic assembler language (BAL) translator utility routine used to assist you in the conversion of your system's source and proc code programs to OS/3 BAL source language. The SCAN routine is not designed to operate on your present system but to operate within the confinement of a 90/30 minimum system configuration.

The purpose of SCAN is to point out and attempt to correct obvious problem areas in your system source and proc programs. It is not designed to be a source level interface between your system and the 90/30. It will not always be possible to assemble and execute the output modules from SCAN, but with the flags and diagnostics from SCAN and the OS/3 assembler, it is possible to convert, with a minimum erfort, your system source and proc modules to OS/3 source and proc format.

To perform BAL translation, SCAN analyzes the source or proc code in order to recognize valid supervisor and data management calls, makes any necessary changes, and flags imcompatible instructions. Proc definitions are altered, where possible, to conform to the OS/3 assembler. Any necessary addressing changes are also provided in order to simulate direct addressing. The actual translation process requires two passes. The first pass alters the appropriate DTFs, replaces or deletes instructions, makes the necessary alterations to procs, determines if the inserted instruction seqeunce should be employed, produces error flags, and builds a symbol table (for I/O areas). The second pass employs the inserted instruction sequence as determined by the first pass, aligns the I/O areas, and, if the user so indicates, prints the listing and outputs the source modules to disc.

C.2.1. Altering Declarative Data Management Calls

The SCAN routine alters, when possible, the DTFs from the your present system format to 90/30 data management format. (Table C—11 lists those DTFs that are recognized by SCAN.) The keyword SAVAREA=SCAN$SAV is added to the DTFs that are recognized, where SCAN$SAV is the label of a half-word-aligned 18-word save area. Reader areas are forced to half-word boundaries by the insertion of a DS OH instruction for DTFs, DTFCR, DTFRP, and DTFRW. Printer I/O areas are generated and the IOA1=SCAN$BUF keyword is added to the DTFs; SCAN$BUF is a half-word-aligned 132-byte storage area for DTFPR and DTFDP.

The keyword parameter WORKA=YES will be added to all 9200/9300 DTFs which use work areas but do not have a keyword parameter to indicate such. These DTFs are: DTFCR, DTFRP, DTFRW, DTFDP, and DTFPT.

If the record format is undefined (RCFM=UNDEF) for a disc DTF (only applicable to DTFDA, DTFRA, or DTFSD), a message is printed in the listing to inform the user that 90/30 data management does not support files with undefined record formats. All DTFs that are not recognized receive a message to inform the user as such.

Any occurrence of a DTFCS will be changed to a comment along with the OPEN and CLOSE imperatives for that file. If the DTF appears as:

```
LABEL  DTFCS  EQFA=END
```

a GET to that file in the form:

```
GET     LABEL,WORK
```

will be replaced by:

```
GETCS   WORK,1,END
C       0,=F'0'
BE      END
```

where:

The compare and branch ins:ructions are inserted to recognize an end-of-file condition.

All accepted disc DTFs having a file type specified as an output file will receive a warning message informing you that data from the file starts at the first position of the I/O area. The target system's I/O buffer begins with an 8-byte count field but on the 90/30, the halfword aligned 8-byte area precedes the I/O buffer. The I/O area keyword must be changed so that the label corresponds to the I/O buffer following the 8-byte count field. The blocksize keyword must be decremented by 8 to abide by the 90/30 requirement of the blocksize being a multiple of the record size. SCAN rounds up the size of the I/O buffer (for all accepted disc DTFs) to the nearest multiple of 256 (to insure device independence within the 90/30), as this is a requisite of all sectored discs. Your file processing code may require alternations in accordance with the aforementioned changes and requirements.

The 90/30 data management accepts all keyword specifications on DTFs. Those that are different spellings of 90/30 keywords will be implemented. All others, such as those that serve no function, will be accepted (not flagged) but ignored. You should inspect the expansion of DTFs and check all PNOTES after assembling. You should be especially careful when using DTFIS due to the numerous differences between the 90/30 and the target systems with respect to that DTF.

Table C—11. DTF Listing

Function	90/30	9200/9300 Recognized
Punch card Serial read/punch Row read/punch	DTFCD	DTFCR DTFRP DTFRW
Printer Drum printer	DTFPR	DTFPR DTFDP
Paper tape Magnetic tape	DTFPT DTFMT	DTFPT DTFMT
Sequential access disc Indexed sequential disc Direct access disc	DTFSD DTFIS DTFDA	DTFSD DTFIS DTFRA DTFDA
Control stream		DTFCS

C.2.2. Altering Supervisor and Imperative Data Management Calls

The SCAN program attempts to replace your system supervisor/data management calls with an equivalent 90/30 call, if the 90/30 system has such a macro. If the call does not appear in the list of macro calls that follow, it is left untouched and when assembled may result in an error. It is up to you to write an equivalent macro or delete the call. (Note that the CNTRL macro is accepted on the 90/30, but the user should verify that the specified parameters serve the intended function).

NOTE:

Relative addressing may cause problems if attempting to branch around imperative macro calls which may expand differently on OS/3 than on target systems.

The following list contains the names of the imperative macro calls that are not identical on the 90/30 and your existing systems. An asterisk next to a name indiates that it is a call on your existing system that has a counterpart on the 90/30 with a different name.

MSG
*CANCL
DSPS2

*CANCL is changed to a comment followed by a flag and insertion of a CANCEL call.

C.2.3. Altering Proc Definitions

The prototype statement and conditional assembly statements of user proc definitions are altered to a form acceptable to the OS/3 assembler. An ampersand (&) is appended to the label and parameters of the proc line if not already present. You must append an ampersand (&) to all positional parameters in the proc definition (i.e. not on the proc line). An equal sign (=) is attached to all keyword parameters on the proc line if not already present. A period (.) is appended to all sequence symbols in the proc definition. These alterations, however, only make the proc definition syntactically correct for the OS/3 assembler, assuming it was syntactically correct on the 9200/9300. SCAN makes no attempt to see that variable symbols are initialized to zeros. If this is not done, these symbols cannot logically or algebraically be applied to a variable symbol with a numeric value. The 9200/9300 local and global symbols of the form L%XX and G%XX are converted to &LXX and &GXX. This may cause a multiply defined symbol. The percent (%) is replaced by a dollar sign if the percent sign does not cause the symbol to be local or glogal (e.g., &L%XX).

C.2.4. Direct Address Simulation

Your system provides for either base/displacement or direct addressing. SCAN simulates direct addressing only for single CSECT modules of 24,576 bytes or less in length. SCAN does not attempt to determine the size of a source module, but maintains a CSECT count. It is your responsibility to ensure that direct addressing is not attempted within a CSECT larger than 24,576 bytes. If direct addressing is being used in multiple CSECT modules or modules larger than 24,576 bytes, SCAN only flags the USING directives with the pseudo registers. (In this case, the results of 90/30 execution of SCAN output are unpredicatable.)

For single CSECT modules less than 24,576 bytes long, direct addressing is simulated by alterting the module as follows:

- Any USINGs with pseudo registers are changed to a comment by the insertion of an asterisk in column 1.

- If there is a leading START or CSECT directive, the instruction sequence (which follows) is inserted directly after the START or CSECT statement. If there is no leading START or CSECT statement, the instruction sequence is inserted directly in the beginning of the source module.

- If there is an END directive with a transfer address on it not equal to the START directive label, that transfer address is inserted in the operand field of the branch instruction of the inserted instruction sequence, and SCAN$FST is inserted in the operand field of the END directive. If there is no END directive or no transfer address on the END directive, or the transfer address equals the START or CSECT label, SCAN$FST is inserted in the operand field of the branch instruction of the inserted instruction sequence.

- The instructions PUNCH 'MOD 4096, 4057' and PUNCH 'MOD 4096, 4056' are inserted before the START or CSECT directive if there is one, or if there are pseudo USINGs used. If there is no START or CSECT directive, the two instructions are inserted directly before the module. These instructions become the first two instructions of the converted module. (These instructions are necessary because registers 0 and 1 cannot be used as base registers due to their use by data management and supervisor macro calls. See the data management user guide, UP-8068 (current version) and the supervisor user guide, UP-8075 (current version).

When stimulating direct addressing, the inserted PUNCH directives cause the first original byte of the module to be loaded at location X'2000'. This leaves X'1FD8' (8152 decimal) bytes of storage unused in front of the module. Any auto-included modules are loaded in this space (i.e., data management modules). It is your responsibility to ensure that the length of the auto-included modules does not exceed X'FD9' (4057 in decimal) bytes, because the two MOD statements otherwise will not perform their intended function. There is a message, following the MOD statements on the link map, that warns you of this problem. If this is the case and you do not take the necessary steps, the user module is not loaded at X'2000', and direct addressing does not work. If the auto-included modules are greater than X'FD9' in length, you must specifically include enough of the auto-included modules to ensure that they are X'FD9' or less bytes in length. Any such specific inclusions are required to occur behind the converted module.

It should be pointed out that on your system, the appropriate DTF calls caused expansion of the processing code as well as the file declaration. On the 90/30, only the file declaration occurs, and external references are generated to cause auto-inclusion of the appropriate IOCS modules. For example, the following IOCS modules require the corresponding amount of space (in bytes): ISAM (4000), SAM (3000), DAM (1000), READER (1800), and PRINTER/PUNCH (1200).

NOTE:

Instructions which reference absolute locations in the supervisor will not be translated or flagged. You must make any necessary changes.

The following code shows the inserted instruction sequence.

```
LABEL     △OPERATION△           OPERAND                                  △
1          10     16
SCAN$XFR   BALR   8,0
           USING  *,8
           LM     2,7,SCAN$LST
           USING  SCAN$FST,2,3,4,5,6,7
           SR     8,8
           DROP   8
           B      [OLD TRANSFER ADDR OR SCAN$FST]
SCAN$LST   DC     A(SCAN$FST)
           DC     A(SCAN$FST+4096)
           DC     A(SCAN$FST+2*4096)
           DC     A(SCAN$FST+3*4096)
           DC     A(SCAN$FST+4*4096)
           DC     A(SCAN$FST+5*4096)
           DC     F'0'
SCAN$FST   EQU    *
```

C.2.5. Conversion of Privileged and Miscellaneous Instructions

The privileged instructions TIO, XIOF, LPSC, SPSC, and SRC do not exist on the 90/30. The SCAN program converts these instructions to comments and produces an EQU* line if any label existed. You must supply the required code to accomplish their function. This should not be a problem because these instructions are privileged and are not normally used by 9200/9300 users. Any HPR instruction will be made into a comment and flagged as being privileged on the 90/30 system. Any instruction that is altered by the SCAN program is flagged in the analytical listing.

On the 9200/9300 card system, the MP (multiply packed decimal), DP (divide packed decimal) and ED (edit) instructions are handled through the external subroutines MPDP or EDIT. Those that use that system must specify the PARAM statement, // PARAM SUB=YES, so that SCAN will perform in the following manner. (Note that all other uses should not use this param statement.) When SCAN encounters either EXTRN MPDP or EXTRN EDIT, the extrn declaration is ignored and a message tells the user as such. Then when the statement BAL 15, MPDP or BAL 15, EDIT is encountered, it is made into a comment (by placing an asterisk in column 1) and a message to indicate as such. If there is a label on the BAL statement, it is removed and placed on the line which follows (the MP, DP or ED instruction). The MP, DP or ED instruction which follows the BAL statement is assembled by the OS/3 assembler, which accepts those op-codes.

The BCW constant is not recognizable so, therefore, any user using PIOCS may have interface problems on OS/3.

NOTE:

In all cases, if column 72 is not blank, the OS/3 assembler expects the following statement to be a continuation and therefore the first 15 columns must be blank.

The OS/3 assembler permits one operand field on ORG directives. If the 9200/9300 user specifies an ORG with 2 operand fields, the ORG will be converted to a comment and a message will follow it.

C.2.6. Input to SCAN

The input to SCAN may be from card, tape, or disc and may be from any number of files. The file names and module names (only the module name for card input) are passed to SCAN by the DISC, SMTAPE, TAPE, or CARD input statement inserted into the data area of the input stream. The operation code for each input statement must specify the character P or S to indicate if the module is a source or proc module. In the case of DISC, TAPE, or SMTAPE input, the modules (in the particular file) to be translated are all considered to be the same type (either source or proc). For example, if some source modules and some proc modules are to be translated from the same file, then two or more input statements are required. An error in the format of an input statement may result in its being ignored. Input can be from all four media in the same job step.

If the input is from cards, the card deck being translated must immediately follow the CARD input statement. A missing CARD input statement results in the card deck being ignored. The card deck being translated is terminated by reaching either a /* or a DISC, TAPE, SMTAPE, or CARD input statement.

The SAM tape will have EBCDIC file processing with standard labels. The record format is fixed length (80 bytes), with blocked records (5 records per block). The first record of each module must be an 80-byte header record. denoted by an X'82' in the first byte of the record. The module name must be in bytes 51—58 of the header record (left justified, blank filled on the right). A module is terminated upon reading the header record for the following module or by an end-of-file condition. A missing header record on the first module of a file terminates processing of that file. Records may be *lost* if the blocking factor is not 5.

On the 9200/9300 library source module tapes, the grouping characteristics are ignored. The user must be careful when using the *gang translation* methods.

The following sample coding shows the format of the input statements:

LABEL 1	ΔOPERATIONΔ 10	OPERAND 16	Δ	COMM
SMTAPE	P S	FILENAME(MODULE1,MODULE2,···MODULEn)		
TAPE	P S	FILENAME(MODULE1,MODULE2,···MODULEn)		
DISC	P S	FILENAME(MODULE1,MODULE2,···MODULEn)		
CARD	P S	MODULENM		

where:

SMTAPE
: Specifies that the file is on SAM tape.

TAPE
: Specifies that the file is on tape in 9200/9300 library source module format.

DISC
: Specifies the file is on disc in OS/3 library source or proc module format.

CARD
: Specifies the module immediately follows on punched cards.

S or P
: Specifies the input and output type as being either source or proc.

FILENAME
: Eight characters representing the file name and must be identical to an LFD name.

MODULENM
: Is one to eight characters representing the module name.

MODULE1—MODULEn
: Are one to eight characters that specify the names of the modules to be converted from this file (FILENAME). If the names of the source modules do not fit on one card, then another card must appear with the same file name, input device type (DISC, TAPE, or SMTAPE), module type (S or P), and the remainder of the module names.

When processing 9200/9300 library tapes, there are a few items you should recognize. First, there will be a message on the console informing you that the tape on device XYZ is not prepped. This message occurs because the tape format does not conform with 90/30 prep standards. The message can be ignored (by keying in I, if a response is required) since it (the error condition) does not effect processing of the tape. Since there is no volume label on the tape, there is no AVR (automatic volume recognition). You must therefore check the console to verify that the tape is assigned to the correct device. If you encounter problems in getting the tape assigned to the correct device, you may have to change the DVC number to conform with the type of device, or set other tape drives down. In either case, the job must be terminated and resubmitted after the necessary changes are made. The VOL statement may require the MCC option depending on the type of drive the tape was built on and the type of drive on which it is to be read. (See the job control user guide, UP-8065 (current version). Following the MCC, (if it is required) there must be a VSN (volume serial number), which should be a dummy name followed by NOV to indicate that the volume checking is suppressed.

For TAPE, SMTAPE, and DISC input only, there are two alternate methods of specifying input. (To serve as an example, TAPE input with SOURCE type will be used, but all combinations are possible):

```
TAPE      S      FILENAME
```

All modules in the specified file are translated.

```
TAPE      S      FILENAME,PREFIX
```

where:

PREFIX
Is one to seven characters.

All modules in the specified file beginning with the given prefix are translated.

C.2.7. Output From SCAN

The SCAN program produces either of two outputs at your request:

- OS/3 librarian format source or proc modules to any OS/3-supported disc.
- An analytical listing on the printer.

The output desired is specified with the PARAM statements (C.2.7.2) LST and OUT. The outputs may be produced individually or in combination.

C.2.7.1. Module Sequencing

Modules processed by SCAN can be sequenced by specifying the proper PARAM statement (C.2.7.2). If the module had sequence numbers, it is resequenced by SCAN. If the sequence field was other than 73—80, the original sequence numbers remain on the source records. This may present no problems for 9200/9300 source modules. All sequencing by SCAN is based on columns 73—80 with an initial value of the first three characters of the module name (or the entire module name if it is less than 3 characters) padded on the right with zeros. There is an increment of 10.

C.2.7.2. Specifying Input/Output Information

There are three PARAM statements that you can use with the SCAN program to specify I/O information: one is used to identify the file into which the output source module or proc module processed by SCAN is placed; one is used to inhibit the analytical listing produced by SCAN; and one is used to request sequencing.

Format:

$$// \textbf{ PARAM OUT=}\left\{\begin{array}{l}\textbf{filename}\\ \textbf{(N)}\end{array}\right\}$$

where:

filename
Is one to eight characters representing the name of the file into which SCAN places its output source or proc modules.

(N)
Specifies no source or macro module output is desired for this job step.

If this PARAM statement is not present in the job control stream, then the default is to put the output modules in $Y$RUN (the run library).

Format:

// **PARAM LST=NOLIST**

where:

NOLIST
Specifies that SCAN is not to prepare an analytical listing.

If omitted, an analytical listing is prepared.

Format:

// **PARAM SEQ=YES**

where:

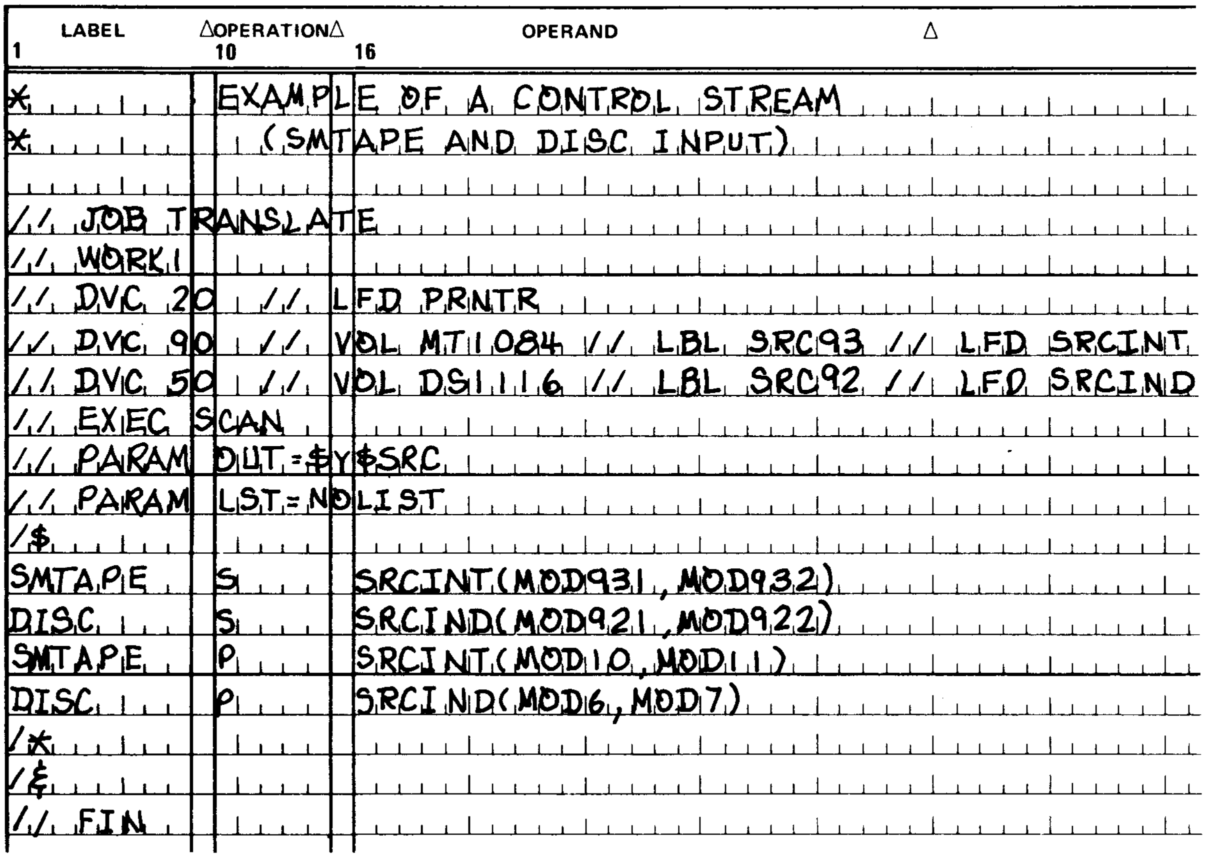
YES

Specifies whether you desire to have the translated modules sequenced.

If omitted, sequencing is not performed.

Typical job control streams that illustrate the coding required for converting tape, disc, and card input modules to SCAN are shown in the examples which follow.

Example 1:



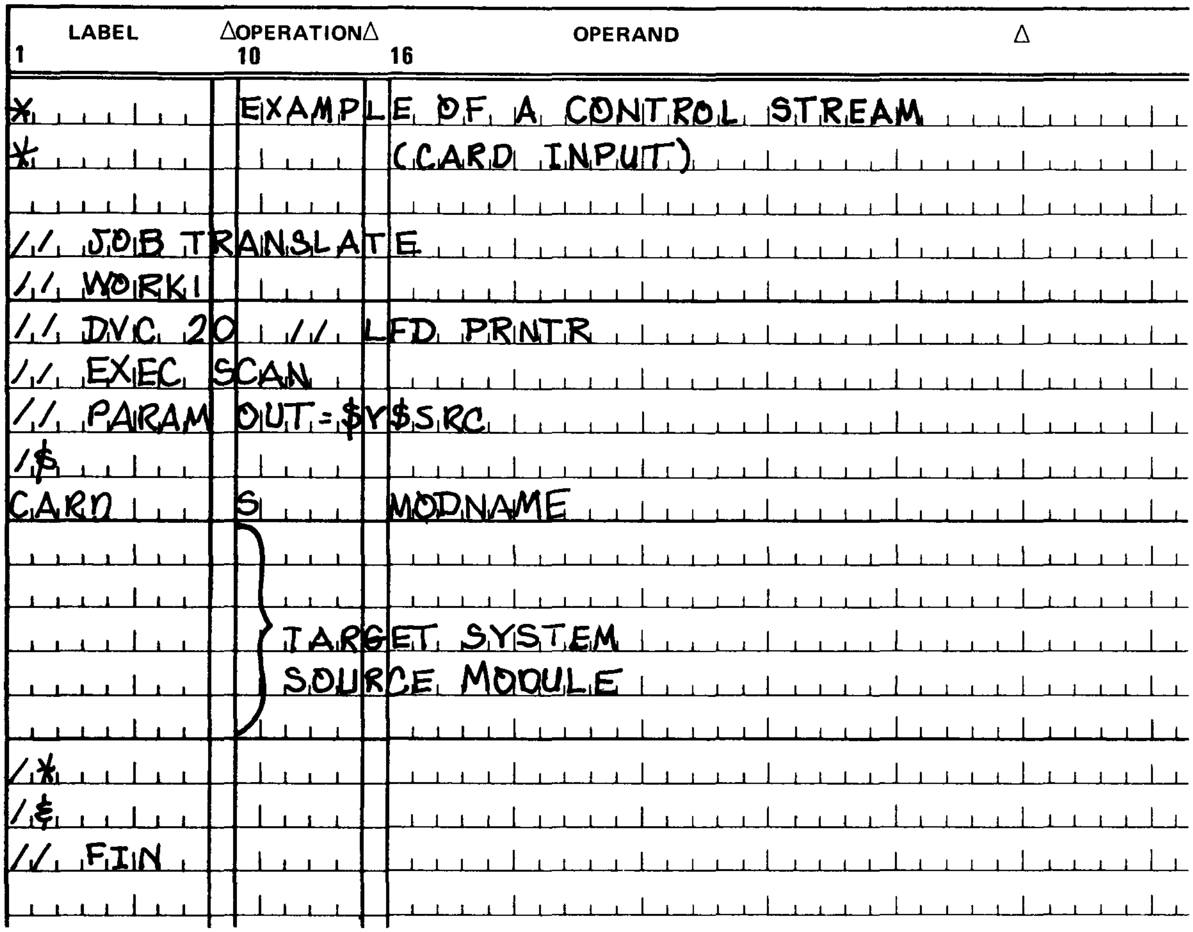
```
LABEL       ΔOPERATIONΔ          OPERAND                    Δ
1            10    16
*        EXAMPLE OF A CONTROL STREAM
*          (SMTAPE AND DISC INPUT)

// JOB TRANSLATE
// WORK1
// DVC 20  // LFD PRNTR
// DVC 90  // VOL MT1084 // LBL SRC93 // LFD SRCINT
// DVC 50  // VOL DS1116 // LBL SRC92 // LFD SRCIND
// EXEC SCAN
// PARAM OUT=$Y$SRC
// PARAM LST=NOLIST
/$
SMTAPE   S     SRCINT(MOD931,MOD932)
DISC     S     SRCIND(MOD921,MOD922)
SMTAPE   P     SRCINT(MOD10,MOD11)
DISC     P     SRCIND(MOD6,MOD7)
/*
/&
// FIN
```

This control stream converts the source modules MOD931 and MOD932 from the SAM tape file SRCINT. The source modules MOD921 and MOD922 are converted from the disc file SRCIND. Next, the proc modules MOD10 and MOD11 are converted from the SAM tape file SRCINT, and then the proc modules MOD6 and MOD7 are converted from the disc file SRCIND. There can be any number of tape and/or disc files used as input to SCAN. If a file cannot be located, an error message is printed along with the TAPE, SMTAPE, or DISC statement that references the file in question. SCAN continues to process the rest of the input records. For disc input, if a module within a file cannot be found, an error message is printed, and the rest of the modules on the input statement are processed.

For TAPE and SMTAPE input, the modules are processed in the order in which they occur on the tape. Following the last translated module, a list of the specified modules that were not found is generated.

Example 2:



```
LABEL      ΔOPERATIONΔ          OPERAND                    Δ
1           10    16
*        EXAMPLE OF A CONTROL STREAM
*              (CARD INPUT)

// JOB TRANSLATE
// WORK1
// DVC 20  // LFD PRNTR
// EXEC SCAN
// PARAM OUT=$Y$SRC
/$
CARD     S     MODNAME
          TARGET SYSTEM
          SOURCE MODULE
/*
/&
// FIN
```

This control stream causes the source module in the card reader to be converted. The output source module is placed in $Y$SRC.

Example 3:

```
LABEL    ΔOPERATIONΔ      OPERAND                     Δ           COMMENTS
1        10    16
*        EXAMPLE OF A CONTROL STREAM
*           (MULTIPLE JOB STEPS)

// JOB TRANSLATE
// DVC 20  // LFD PRNTR
// DVC 90  // VOL MT104  // LBL SRC93  // LFD SRCINT
// DVC 91  // VOL MT105  // LBL PRC93  // LFD PRCINT
// DVC 50  // VOL DS1126 // LBL SRC92  // LFD SRCIND
// DVC 51  // VOL DS1127 // LBL PRC92  // LFD PRCIND
// WORK1
// OPTION REPEAT
// EXEC SCAN
// PARAM OUT=$Y$SRC
/$
TAPE     S     SRCINT,XYZ$
TAPE     S     SRCINT,(A,B,C)
CARD     S     MODULE1
               }
               }  SOURCE MODULE
               }  (MODULE1)
               }
DISC     S     SRCIND,WXY
DISC     S     SRCIND(X,Y)
/*
// PARAM OUT=$Y$MAC
/$
TAPE     P     PRCINT,SYS
TAPE     P     PRCINT(DRC1,DRC2)
DISC     P     PRCIND
CARD     P     PRCMOD1
               }
               }  PROC MODULE
               }   (PRCMOD1)
               }
/*
/&
// FIN
```

C.2.8. Related Software Components and Interfaces

The SCAN program requires and interfaces with the OS/3 data management and the following library utility subroutines:

LU$EAT

SRC$READ

SRC$GEN

DRC$ALL

C.2.9. Data Base

SCAN is capable of accepting input from any of four mediums:

- 80-column cards
- Source modules in OS/3 source or proc format from any OS/3 disc supported library, as created by the OS/3 librarian
- 80-byte card images on compatible SAM tape
- Source modules in 92/9300 source format from any 92/9300 tape supported library, as created by 92/9300 librarian

C.2.10. Error Processing

The error messages that SCAN puts out are listed and described in the OS/3 system messages manual, UP-8076 (current version). If you, as a SCAN user, receive any data management or job control error messages (either on the console or printer), you must refer to the current versions of the appropriate reference manuals (data management user guide, UP-8068 and job control user guide, UP-8065). The SCAN error messages that cancel the job may have accompanying data management or job control error messages which can be of great aid to you. (Note: All SCAN error messages are displayed on the printer.)

There is no *bypass* routine that allows you to skip over blocks of records (in which there was a read error) and continue processing. Since a good deal of the translation is based on previously encountered statements, incomplete processing (i.e., skipping over blocks) can result in an incorrect translation.

C.3. CONVERTING YOUR EMULATED 8410 DISC FILES TO OS/3 DATA FILES BY DCON92 PROGRAM

The DCON92 program assists you in converting your emulated 8410 disc files to an OS/3 data file without the need of transferring the data through an intermediate 9200/9300 medium such as card, tape, or 8411/8414 disc. This conversion utility program accepts as input your emulated 8410 SAM, DAM, and ISAM files and converts these files to an OS/3 SAM, DAM, or ISAM file as per your instructions. These instructions are submitted as part of the run stream to DCON92 in the form of parameter specifications. The specifications describe the input/output files and provides DCON92 with information required for unloading and reloading your present 8410 files to an OS/3 disc. DCON92, which is actually a job filed in the job control stream library file ($Y$JCS) of the OS/3 SYSRES pack, accepts the parameter specifications and initates a JPROC referred to as DCON. The specifications are processed by DCON and the run stream is expanded with variable information necessary to execute the two job steps DCON92 must perform for the conversion.

The first job step is to unload your emulated 8410 file. DCON92 generates the control streams for emulating the 9200/9300 operating system and for running the disc file UNLOAD program under emulation. This allows you to unload your existing file in a card image form (80 bytes) which is spooled by the 90/30 system.

The second job step is to format and reload your file onto an OS/3 disc. DCON92 provides the control stream for executing the disc file RELOAD program which obtains the spooled card images and creates the OS/3 SAM, DAM, or ISAM file according to your specifications.

C.3.1. Interfaces Required to Perform File Conversion

The hardware requirements you must meet to perform file conversion are a 90/30 system with a minimum main storage of 49K, two 90/30 discs (8411, 8414, 8416, 8418, or 8430), a card reader, and a printer.

The OS/3 SYSRES pack must be mounted as well as the OS/3 disc packs containing the emulated input file and the OS/3 output file. (See Figure C—6.)

Your only remaining requirement is to prepare the control input containing the input and output data file specifications to be submitted to the DCON92 program.

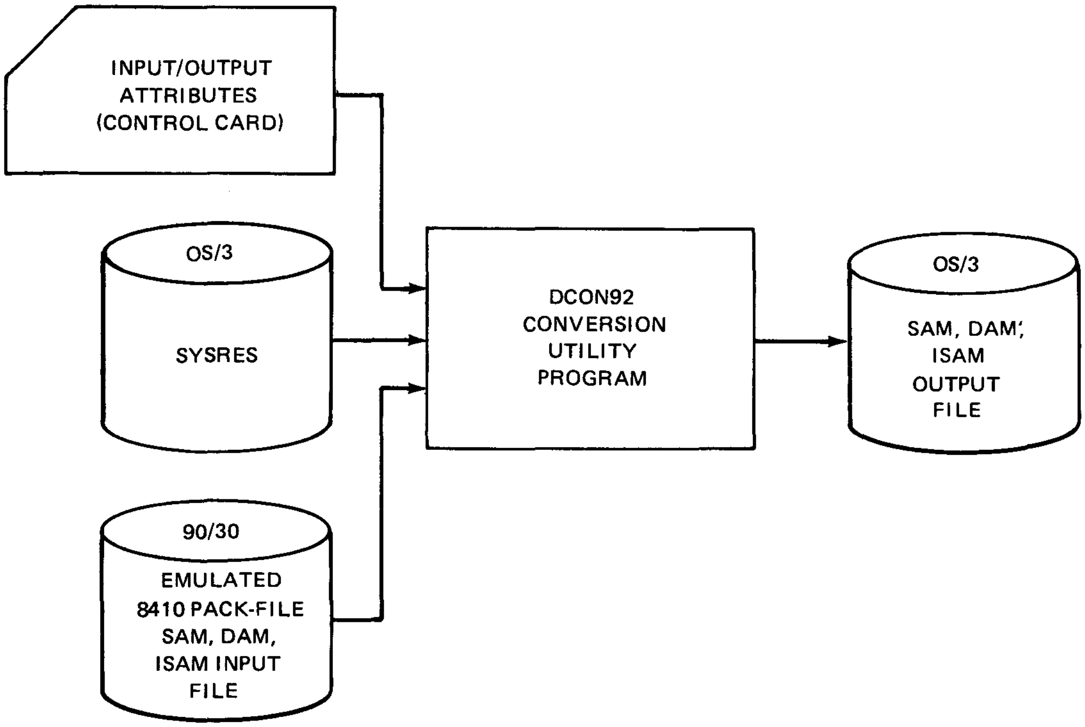

INPUT/OUTPUT ATTRIBUTES (CONTROL CARD)
OS/3
SYSRES
90/30
EMULATED 8410 PACK-FILE SAM, DAM, ISAM INPUT FILE
DCON92 CONVERSION UTILITY PROGRAM
OS/3
SAM, DAM', ISAM OUTPUT FILE

Figure C—6. Input/Output Interface Requirements for DCON92 Conversion Utility Program

C.3.2. Input Requirements to DCON92

Your input to DCON92 consists of groups of control cards; the // DCON control cards, the file description card for the disc file UNLOAD program, and the PARAM card for the disc file RELOAD program. The sequence in which they appear in the control stream is as shown in Figure C—7.

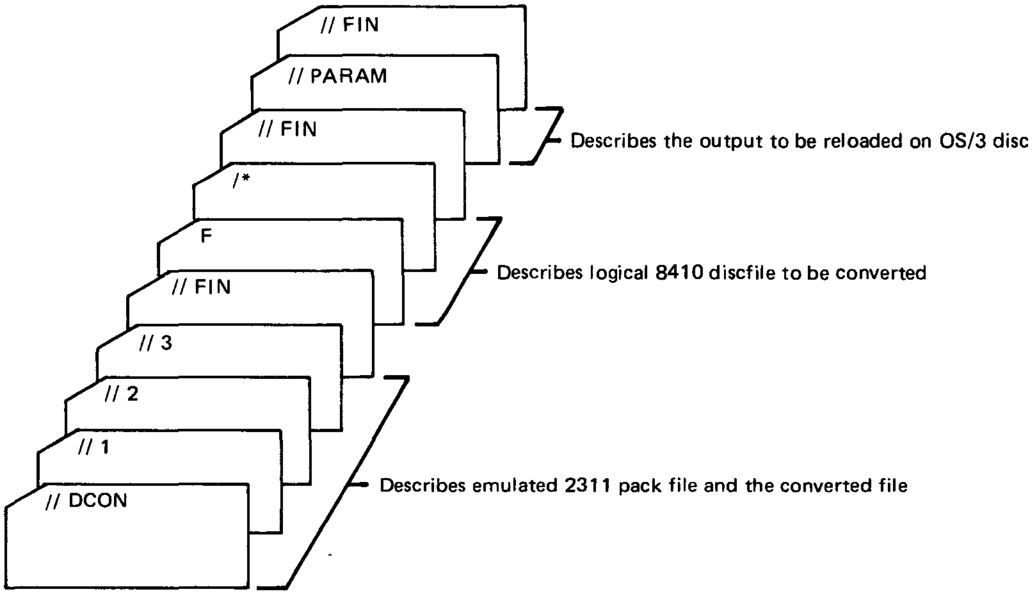

// FIN
// PARAM
// FIN
/*
F
// FIN
// 3
// 2
// 1
// DCON
Describes the output to be reloaded on OS/3 disc
Describes logical 8410 discfile to be converted
Describes emulated 2311 pack file and the converted file

Figure C—7. Input Card Sequence to DCON92 Conversion Utility Program

C.3.2.1. DCON Control Card Preparation

The keyword parameters prefixed with the letter U form a group which describes an emulated 8410 pack-file image. Such a group of keywords must be specified for each emulated 8410 pack-file which is a volume of the 8410 file. Only one file may be described.

The decimal number n must be the same for each keyword of the group and different for each group (volume) described.

The keyword parameters prefixed with the letter R and ending with the suffix r form a group which describes the converted file. Such a group must be specified for each volume of the converted file.

The decimal number r in the keyword RVOLr must be 1 to 5 and different for each (OS/3 volume) described. For output multivolume files, all volumes will be assigned by DCON92 to the same 90/30 drive.

Format:

LABEL	ΔOPERATIONΔ	OPERAND

//	DCON	LLDVCn=ddd
		ULBLn=llllllll
		UPFMNTn=00–99
		UPFPUn=p
		UVOLn=vvvvvv
		SDVC=sss
		SVOL=iiiiii
		RDVC=ddd
		RENOCYLr=ccc
		RETYPE= $\begin{Bmatrix}\text{DA}\\\text{IS}\\\text{SQ}\end{Bmatrix}$
		RLBL=output-filelabel
		RVOLr=vvvvvv
		$\left[\text{DATEF=}\begin{Bmatrix}2\\3\\4\end{Bmatrix}\right]$
		[,RERUN=YES]

UDVCn Keyword Parameter:

UDVCn=ddd
A decimal number specifying the 90/30 device number for the OS/3 disc pack containing the emulated 8410 volume file.

ULBLn Keyword Parameter:

ULBLn=llllllll
Specified the OS/3 file label of the emulated 8410 disc pack. Up to eight alphanumeric characters permitted.

UPFMNTn Keyword Parameter:

UPFMNTn=00–99
Decimal number specifying the order in which the OS/3 volumes are to be mounted. If all volumes of the file are to be mounted initially, specify 00 for every keyword group in the file.

UPFPUn Keyword Parameter:

UPFPUn=p
Specifies the 9200/9300 physical unit number.

UVOLn Keyword Parameter:

UVOLn=vvvvvv
Specifies the volume serial number of the OS/3 disc pack containing the 8410 pack-file.

SDVC Keyword Parameter:

SDVC=sss
A decimal number specifying the 90/30 device number for the OS/3 disc pack containing the SYSIOPI pack-file containing UNLOAD10.

SVOL Keyword Parameter:

SVOL=iiiiii
Specifies the volume serial number of the OS/3 disc pack containing the SYS10PI pack-file containing UNLOAD10.

RDVC Keyword Parameter:

RDVC=ddd
A decimal number specifying the OS/3 disc unit containing the volume of the output file.

RENOCYL Keyword Parameter:

RENOCYL=ccc
Specifies the number of cylinders to be allocated to this volume of the output file.

RETYPE Keyword Parameter:

$$\textbf{RETYPE=}\left\{\begin{array}{l}\textbf{DA}\\\textbf{IS}\\\textbf{SQ}\end{array}\right\}$$

Alphabetic characters specifying the type of output file desired.

DA
Direct access file.

IS
Indexed sequential file.

SQ
Sequential file.

RLBL Keyword Parameter:

RLBL=output-filelabel
Specifies the label of the output file up to 44 characters.

RVOLr Keyword Parameter:

RVOLr=vvvvvv
Specifies the volume serial number of the OS/3 output volume.

DATEF Keyword Parameter:

$$\textbf{DATEF=}\left\{\begin{array}{l}\textbf{2}\\\textbf{3}\\\textbf{4}\end{array}\right\}$$

Establishes the date format for the 8410 file creation and expiration dates.

2
Format is mmddyy.

3
Format is ddmmyy.

4
Format is yyddd.

If omitted, format is assumed as yymmdd.

RERUN Keyword Parameter:

RERUN=YES

Specifies that the output file is to be deallocated prior to another execution of the same DCON92 run. This parameter must be specified if DCON92 is to be rerun after // EXEC RELOAD appears on the display console. The display occurs after the output file has been allocated. It is necessary during a subsequent rerun with the same output file to deallocate the file.

C.3.2.2. File Description Card Preparation

The file description card describes the logical 8410 file volume to be converted. It provides the information used by the disc file UNLOAD program to load and modify the appropriate disc I/O processing routines for unloading your 8410 disc file images. A file description card is required for the file to be converted. The format for the file description card is presented in Table C—12.

Table C–12. File Description Card Format (Part 1 of 2)

Card Column	Specification	Description
1–12	F01050000000	Identifies the card type
13–14	{ FU FB VU VB UN IG EN } blank	Specifies the input file record format FU – Fixed-length unblocked FB – Fixed-length blocked VU – Variable-length unblocked VB – Variable-length blocked UN – Unblocked IG – Ignore. Do not copy records consisting of binary 0. EN – Cease copying the file if a record of binary 0 is encountered. Blank – Copy entire file until end-of-file is detected. NOTE: The IG, EN, and blank specifications may only be used if file organization is direct and record format is assumed to be fixed-length unblocked.
15	[3]	Specifies that the input file is an 8410 ISAM file created by the assembler language program using the optional 9200/9300 stagger effect (PROC=9300 specified on DTFIA)
16–19	wwww	Record size in decimal

Table C–12. File Description Card Format (Part 2 of 2)

Card Column	Specification	Description
21–24	kkkk	Block size in decimal – Must be a multiple of record size – May be larger than existing input file record in order to cause reblocking of output file. Applicable only to transcribing a fixed-length blocked file due to peculiarities of the 9200/9300 ICOS routines
26–28	eee	For ISAM only, key length in decimal
30–33	bbbb	For ISAM only, key location relative to the beginning of record with 0000 signifying the first byte of the record
35	I S D	File organization
36	blank 1–8	For ISAM only, the number of volumes comprising the file, with the blank assumed to mean 1. All volumes must be online while executing DCON. DCON assumes that volume of an ISAM file will be mounted on consecutive logical units.
37–80	a_1,a_2,a_{44}	File identification; i.e., the 8-byte character string which uniquely identifies the file to be copied Although only the first eight characters are used for locating an 8410 input file, additional characters may be appended to expand the file ID on the output medium. See RLBLn keyword.

NOTE:

A blank occuring in a decimal field of the F card is treated as a zero.

C.3.2.3. RELOAD PARAM Card Preparation

The PARAM card informs the RELOAD program of the attributes of the output file being produced. This card need only be specified if the output file type (DAM, SAM, or ISAM) differs from the input file being converted.

Format:

LABEL	△OPERATION△	OPERAND
//	PARAM	[BLKL=n] $\left[\text{ERRCOUNT=}\begin{Bmatrix}0\text{–}32767\\ \mathbf{0}\end{Bmatrix}\right]$ $\left[\text{FTYP=}\begin{Bmatrix}\text{D}\\ \text{I}\\ \text{S}\end{Bmatrix}\right]$ [KLOC=n] [KSIZ=3–253 $\left[\text{OVFL=}\begin{Bmatrix}0\text{–}80\\ \mathbf{20}\end{Bmatrix}\right]$ [RECL=n] $\left[\text{RTYP=}\begin{Bmatrix}\text{FB}\\ \text{FU}\\ \text{VB}\\ \text{VU}\end{Bmatrix}\right]$

BLKL Keyword Parameter:

BLKL=n

A decimal number specifying the block length of the output file. It does not, however, include the 8-byte count field. It allows you to have an output block size that is different from the input block size. For fixed records in the output file, the block and record size (whether specified by PARAM card or assigned by default) must maintain the following relationship:

1. b = r DAM
2. b = nr SAM
3. b = n(r+5)+2 ISAM

where b is blocksize, r is record size, and n is the blocking factor. (See OS/3 data management user guide, UP-8068 (current version) for details.)

If omitted, the output blocksize is assumed to be the same as the input blocksize unless the following conditions exist.

1. The input file is an ISAM file with fixed records.
2. The output file is an ISAM file with fixed records (whether coded on PARAM cards or assigned by default).
3. Both BLKL and RECL parameters are not specified.

In these cases, DCON92 assigns default values determined by the following:

1. Output record size is assumed equal to the input record size. (See RECL parameter.)

2. The input blocking factor is used to compute the default output blocksize so that the special OS/3 block size/record size relationship holds.

This facility allows you to convert your ISAM without supplying any PARAM cards to DCON92.

ERRCOUNT Keyword Parameter:

$$\textbf{ERRCOUNT=}\left\{\begin{matrix}\textbf{0–32767}\\ \textbf{0}\end{matrix}\right\}$$

For a tape intermediate file, the value specified determines the number of nonfatal errors that can be tolerated during file processing. When a nonfatal error occurs during tape read, the offending block is shipped, and a new block is read (all records in your input file, therefore, may not appear in your output file).

For a card intermediate file, the value specified determines the number of sequence check errors that can be ignored during card processing. The contents of out-of-sequence cards are processed, however. Refer to the section on error processing (C.1.2.6) for a discussion of nonfatal errors. All errors encountered during directory read are considered fatal and result in the termination of file conversion.

If omitted, zero is assumed and the first nonfatal error encountered causes file processing to be suspended and the file closed.

FTYP Keyword Parameter:

$$\textbf{FTYP=}\left\{\begin{matrix}\textbf{D}\\ \textbf{I}\\ \textbf{S}\end{matrix}\right\}$$

Alphabetic characters specifying the type of output file being created.

D

Creates a direct access (DAM) output file.

I

Creates an index sequential (ISAM) output file.

S

Creates a sequential (SAM) output file.

If omitted, the output file type is the same as the input file.

KLOC Keyword Parameter:

KLOC=n

A decimal number specifying the key locations for an ISAM output file. The value specified represents the position of the key field from the beginning of the record, starting from zero. It includes the two bytes of the record descriptor for variable-length records.

If omitted, the key size is obtained from the input key location. If the input file is not an ISAM file, a value of zero is assumed for fixed-length records and 2 for variable-length records.

KSIZ Keyword Parameter:

KSIZ=3—253

Specifies the key size for an ISAM output file.

If omitted, the key size is assumed to be the same as the input file, provided that the input file is an ISAM file. If the key size of the input file is unobtainable, DCON92 terminates.

OVFL Keyword Parameter:

$$\textbf{OVFL=}\left\{\begin{matrix}\textbf{0–80}\\ \textbf{20}\end{matrix}\right\}$$

Specifies the percentage of disc space to be used as an overflow area for an ISAM output file.

RECL Keyword Parameter:

RECL=n

A decimal number specifying the output record length for fixed-length records. If the output record size is larger than input records (variable or undefined length to fixed-length conversion) or larger than all input records (variable, undefined, or fixed-length conversion). DCON92 pads the output records with binary 0's. If the output record size is smaller than any input record, DCON92 terminates as soon as such a condition is encountered.

If omitted, the record length is assumed to be the same as the input record length provided that the input file is a fixed-length file.

For DAM output files, the record size is assumed to be the same as the input record size provided that the input records are fixed-length. If the input file has variable-length records, the output record length is unavailable, and DCON terminates.

RTYP Keyword Parameter:

$$\text{RTYP=}\begin{Bmatrix}\text{FB}\\ \text{FU}\\ \text{VB}\\ \text{VU}\end{Bmatrix}$$

Defines the output file record type. This keyword must be specified if the input file record type is undefined.

FB

Specifies fixed, blocked records.

FU

Specifies fixed, unblocked records.

VB

Specifies variable, blocked records.

VU

Specifies variable, unblocked records.

Parameters FB and VB may not be specified for a DAM output file, and FU and VU may not be specified for an ISAM output file.

If omitted, the output file record type is assumed to be the same as the input file record type.

C.3.3. Data Bases Required for Conversion

Input data must be contained in an 8410 file located in one or more 8410 pack images emulated on an OS/3 pack image file.

The 8410 file may be a 9200/9300 SAM, DAM, or ISAM file. If the 9200/9300 file is a multivolume file, the other volumes may occupy OS/3 pack-image files that are on the same or other OS/3 disc packs of the same type device.

Output data will be an OS/3 SAM, DAM, or ISAM file contained on 90/30 discs. Output is via SAT, and the rules of SAM apply. The output file format, such as blocking factor, record size, etc., may be varied between input and output as per your specified parameters.

C.3.4. Initiating the DCON92 Program

To begin the conversion, a supervisor with output spooling must be available, and the operator must place the input cards in the control stream reader and key in RU DCON92 from the console.

C.3.5. Console Messages

Console messages can occur during input file read under 9200/9300 emulation. In some cases, the operator must respond with a specific action. The messages, their probable cause, and the operator action required are presented in the OS/3 system messages manual, UP-8076 (current version).

Error messages may also be displayed on the console. These messages will be displayed during the building of the output file (after // EXEC RELOAD appears on the console). These messages and an explanation of their probable cause are also presented in UP-8076.

C.3.6. Example of Input Control Stream for DCON92

```
   LABEL     ΔOPERATIONΔ            OPERAND                     Δ           COMMENTS
1            10     16                                                                          72
// DCON   UDVC1#051,UVOL1#PK9030,ULBL1#PACKA1,UPFMNT1#00,DATEF#2,                               X
// 1      UDVC2#051,UVOL2#PK9030,ULBL2,PACKA2,UPFMNT2#00,                                       X
// 2      RDVC1#052,RVOL1#123456,RETYPE1#IS,RENOCYL1#4,RLBL1#FILEONE
// FIN
F0002PF8410 FB 0080 0160 004 0000 I2FILEONE
/*
// FIN
 (RELOAD  PARAMETERS)
// FIN
```

C.3.7. Restrictions in the Use of DCON92

Only one file per run can be converted; however, that file is taken through the entire process of conversion from 9200/9300 8410 file format to OS/3 compatible disc file format. Multiple use of DCON provides an uncomplicated method for converting multiple files without loss of processing time.

Appendix D. Device Codes

Table D—1 lists the device codes (DVC) used on the emulator descriptor cards. These device codes are identical to the logical unit numbers used on OS/3 control statements.

Table D–1. Emulator Device Codes (Part 1 of 3)

Device Code (DVC)	Device Type and Features
1	Reader of 0920 paper tape subsystem
2	Reader of 0920 paper tape subsystem
3	Punch of 0920 paper tape subsystem
4	Punch of 0920 paper tape subsystem
5	2703 optical document reader
6	2703 optical document reader
7	Spare
8	Spare
9	Spare
10	Spare
11	Spare
12	Spare
13	Spare
14	Spare
15	Spare
16	Spare
17	Spare
18	Spare
19	Spare
20	Any printer, no features specified
21	Any printer, no features specified
22	0773 printer subsystem, no optional features
23	0773 printer subsystem, no optional features
24	0776 printer subsystem, no optional features
25	0776 printer subsystem, no optional features
26	0768 printer subsystem, no optional features
27	0768 printer subsystem, no optional features
28	0770 printer subsystem, no optional features
29	0770 printer subsystem, no optional features
30	Any card reader subsystem, no features specified
31	Any card reader subsystem, no features specified
32	0717 card reader subsystem, no features specified
33	0717 card reader subsystem, no features specified
34	0716 card reader subsystem, no features specified
35	0716 card reader subsystem, no features specified

Table D–1. Emulator Device Codes (Part 2 of 3)

Device Code (DVC)	Device Type and Features
36	Reserved
37	Reserved
38	Spare
39	Spare
40	Any card punch subsystem, no features specified
41	Any card punch subsystem, no features specified
42	0605 card punch subsystem, no features specified
43	0605 card punch subsystem, no features specified
44	0604 card punch subsystem, no features specified
45	0604 card punch subsystem, no features specified
46	Spare
47	Spare
48	Spare
49	Spare
50	Any disc
51	Any disc
52	Any disc
53	Any disc
54	Any disc
55	Any disc
56	Any disc
57	Any disc
58	Any disc
59	Any disc
60	8416 disc subsystem
61	8416 disc subsystem
62	8416 disc subsystem
63	8416 disc subsystem
64	8418 disc subsystem (Mod-I)
65	8418 disc subsystem (Mod-I)
66	8418 disc subsystem (Mod-I)
67	8418 disc subsystem (Mod-II)
68	8418 disc subsystem (Mod-II)
69	8418 disc subsystem (Mod-II)
70	8430 disc subsystem
71	8430 disc subsystem
72	8430 disc subsystem
73	8430 disc subsystem
74	8430 disc subsystem
75	8430 disc subsystem
76	8430 disc subsystem
77	8430 disc subsystem
78	8430 disc subsystem
79	8430 disc subsystem
80	8414 disc subsystem
81	8414 disc subsystem
82	8414 disc subsystem
83	8414 disc subsystem
84	8414 disc subsystem
85	8414 disc subsystem

Table D–1. Emulator Device Codes (Part 3 of 3)

Device Code (DVC)	Device Type and Features
86	8411 disc subsystem
87	8411 disc subsystem
88	8411 disc subsystem
89	8411 disc subsystem
90	Any tape, no features specified
91	Any tape, no features specified
92	Any tape, no features specified
93	Any tape, no features specified
94	Any tape, no features specified
95	Any tape, no features specified
96	Any tape, no features specified
97	Any tape, no features specified
98	Any tape, no features specified
99	Any tape, no features specified
100	Any tape, 9-track phase encoded
101	Any tape, 9-track phase encoded
102	Any tape, 9-track phase encoded
103	Any tape, 9-track NRZI
104	Any tape, 9-track NRZI
105	Any tape, 9-track NRZI
106	Any tape, 7-track NRZI
107	Any tape, 7-track NRZI
108	Any tape, 7-track NRZI
109	Any tape, 7-track NRZI
110	Slow tape, 9-track phase encoded
111	Slow tape, 9-track phase encoded
112	Slow tape, 9-track phase encoded
113	Slow tape, 9-track NRZI
114	Slow tape, 9-track NRZI
115	Slow tape, 9-track NRZI
116	Slow tape, 7-track NRZI
117	Slow tape, 7-track NRZI
118	Slow tape, 7-track NRZI
119	Slow tape, 7-track NRZI
120	Fast tape, 9-track phase encoded
121	Fast tape, 9-track phase encoded
122	Fast tape, 9-track phase encoded
123	Fast tape, 9-track NRZI
124	Fast tape, 9-track NRZI
125	Fast tape, 9-track NRZI
126	Fast tape, 7-track NRZI
127	Fast tape, 7-track NRZI

Index

Term	Reference	Page
A		
Accessing methods, 90/30 under emulation	3.2	3—1
Advisory messages, emulator	B.4.3	B—15
B		
BAL conversion, SCAN routine	4.3.1	4—3
	C.2	C—41
BAL translation		
declarative data management calls	C.2.1	C—41
direct addressing	C.2.4	C—44
DTF listing	Table C—11	C—43
imperative data management calls	C.2.2	C—43
privileged instructions	C.2.5	C—46
proc definitions	C.2.3	C—43
source code analyzer (SCAN) routine	C.2	C—41
supervisor calls	C.2.2	C—43
Buffering specifications		
for BAR printer	6.2	6—21
for 0603 punch	6.2	6—22
for 0711 reader	6.2	6—12
C		
Card punch types required by emulated program		
diagnostics	6.2	6—23
how to specify	6.2	6—23
Card reader types required by emulated program		
diagnostics	6.2	6—10
how to specify	6.2	6—10
Card type A identification		
diagnostics	6.2	6—28
how to specify	6.2	6—28
Card type B identification		
diagnostics	6.3	6—37
how to specify	6.3	6—37
Card type C identification		
diagnostics	6.4	6—40
how to specify	6.4	6—40
Card type D identification		
diagnostics	6.5	6—47
how to specify	6.5	6—47
Card type E identification		
diagnostics	6.6	6—54
how to specify	6.6	6—54
Card type F identification		
diagnostics	6.7	6—65
how to specify	6.7	6—65
Cards, control	See control cards.	
Channel punching, given line number		
diagnostics	6.6	6—51
how to specify	6.6	6—51
Character positioning, maximum line length required		
diagnostics	6.2	6—19
how to specify	6.2	6—19
COBOL conversion		
compilation differences, OS/3, 9200/9300	4.3.3	4—5
OS/3 provided compilers	4.3.3	4—5
Codes, device	Appendix D	
Command control word chaining	6.2	6—13
Command formats	See commands/ messages, emuiation.	
Commands, alter display	B.4.4.2	B—16

Term	Reference	Page
Commands/messages, emulation		
category	B.4	B—3
descriptions	B.4.1	B—4
formats	B.4.1	B—4
program directives	B.4.1	B—4
	B.4.4.1	B—15
Console specifications, system	B.3	B—2
Control cards, PIMAGE program		
conventions	A.1.2.1	A—7
preparation	A.1.2.1	A—7
Control storage load procedure, 9200/9300 emulator	B.1	B—1
Conventions, PIMAGE program	A.1.2.1	A—7
Conversion mechanism, disc file RELOAD program	C.1.2.1	C—18
Conversion utilities		
BAL translation, SCAN routine	C.2	C—41
disc file conversion, 9200/9300 to 90/30	C.1	C—1
	C.2	C—41
	Fig. C—1	C—1
Current job name		
diagnostics	6.7	6—56
how to define	6.7	6—56

D

Descriptor card preparation	Section 6	
	Fig. 5—2	5—4
Descriptor card type A		
definition	6.2	6—4
diagnostics	6.2	6—5
field functions and specifications	6.2	6—5
Descriptor card type B		
definition	6.3	6—29
diagnostics	6.3	6—30
field functions and specifications	6.3	6—30
Descriptor card type C		
definition	6.4	6—38
diagnostics	6.4	6—39
field functions and specifications	6.4	6—39
Descriptor card type D		
definition	6.5	6—41
diagnostics	6.5	6—42
field functions and specifications	6.5	6—42

Term	Reference	Page
Descriptor card type E		
definitions	6.6	6—48
diagnostics	6.6	6—49
field functions and specifications	6.6	6—49
Descriptor card type F		
definitions	6.7	6—55
diagnostics	6.7	6—56
field functions and specifications	6.7	6—56
Descriptor cards, emulator generation		
control stream sequencing	Fig. 6—1	6—3
function	Table 6—1	6—2
preparation	5.3	5—2
	6.1	6—1
	Fig. 5—2	5—4
type A	6.2	6—4
type B	6.3	6—29
type C	6.4	6—38
type D	6.5	6—41
type E	6.6	6—48
type F	6.7	6—55
types	Table 6—1	6—2
Device assignment, conservation	7.6	7—2
Device channel address association, 9200/9300, 90/30		
diagnostics	6.2	6—27
how to specify	6.2	6—27
Device code, 90/30 disc drive for system or data files	See logical unit number, 90/30 disc drive for system or data files.	
Device codes	Appendix D	
Diagnostic messages		
descriptor card type A	6.2	6—5
descriptor card type B	6.3	6—30
descriptor card type C	6.4	6—39
descriptor card type D	6.5	6—42
descriptor card type E	6.6	6—49
descriptor card type F	6.7	6—56
Disc copying (COPY$10) program	A.2	A—17
Disc file conversion		
BAL translation by SCAN routine	C.2	C—41
9200/9300 to 90/30	C.1	C—1
	Fig. C—1	C—1
DCON92 program	C.3	C—55
utilities required	Appendix C	
Disc file RELOAD program	See RELOAD program.	

Term	Reference	Page
Disc file UNLOAD program	See UNLOAD program.	
Disc pack identification, 8410		
diagnostics	6.3	6—36
how to specify	6.3	6—36
Disc prep, emulation considerations	7.5	7—2
Display/alter commands	B.4.4.2	B—16
Display only, EMULAT phase limitation		
diagnostics	6.7	6—64
how to specify	6.7	6—64
Dump limit card		
format	Table A—2	A—5
function	A.1.1.2	A—5
stream sequence	Fig. A—2	A—4
Dump, 8410 disc		
control card formats	Table A—1	A—5
	Table A—2	A—5
	Table A—3	A—7
control card sequence	A.1.1.1	A—4
	Fig. A—2	A—4
control cards required	A.1.1	A—4

E

Emulation		
basic approach for use	3.2	3—1
how it works	1.1	1—1
	Fig. 1—2	1—5
introduction	1.1	1—1
logical approach to job processing	Fig. 1—1	1—4
peripheral counterparts	Table 1—1	1—2
what to consider prior to use	2.1	2—1
what will and will not be emulated	1.3	1—6
Emulation aids, 8410 disc packs		
description	A.1	A—1
DUMP/RESTORE routine	A.1.2	A—7
overview	Fig. A—1	A—2
PIMAGE routine	A.1	A—1
programs to transcribe data files	A.1	A—1
restoration to 90/30 packs	A.1.2	A—7
Emulation operating instructions, 9200/9300		
commands/messages	B.4	B—3
emulator program loading	B.2	B—2
initial loading	B.1	B—1
system console specifications	B.3	B—2
Emulator advisory messages	B.4.3	B—15
Emulator creation, system generation	2.1.3	2—4
	5.3	5—2
	Fig. 5—1	5—3

Term	Reference	Page
Emulator generation		
aids, image transfer disc subsystem	Appendix A	
basic approach	3.2	3—1
conditions determining emulator number	5.2	5—1
creation	2.1.3	2—4
	5.3	5—2
	Fig. 5—1	5—3
data file consideration	2.1.2	2—3
descriptor card function	5.3	5—2
	6.1	6—1
	Table 6—1	6—2
descriptor cards required	6.1	6—1
	Fig. 6—1	6—3
file conversion	4.3	4—3
introduction	Section 5	
load time considerations	Fig. 3—2	3—6
main storage requirements	2.1	2—1
	Table 2—1	2—1
	Table 2—2	2—2
methods of accessing 90/30	3.2	3—1
planning	Section 2	
process	5.3	5—2
	Fig. 5—1	5—3
program conversion	4.3	4—3
sequencing order, descriptor cards	Fig. 6—1	6—3
system preparation	3.1	3—1
system transfer	Fig. 3—1	3—5
tailoring the emulator program	3.3	3—2
the EMULAT phase	5.1	5—1
what to consider before executing under emulation	3.2	3—1
	3.3	3—2
	Table 3—1	3—4
	Fig. 3—1	3—5
Emulator hang	7.8	7—2
Emulator main storage requirements		
defining	7.2	7—1
program execution	7.2	7—1
specification on type F card	6.7	6—61
Emulator name		
diagnostics	6.2	6—5
how to specify	6.2	6—5
Emulator only, EMULAT phase limitation		
diagnostics	6.7	6—64
how to specify	6.7	6—64
Emulator program looping, minimizing methods	7.2	7—2
Emulator program tailoring	3.3	3—2
	Fig. 3—1	3—5
	Fig. 3—2	3—6
	Table 3—1	3—4
Emulator size, main storage requirements		
diagnostics	6.7	6—61
how to specify	6.7	6—61

Term	Reference	Page
END card		
format	Table A—3	A—7
function	A.1.1.3	A—7
stream sequence	Fig. A—2	A—4
Error codes, emulator	B.5	B—16
Error messages, UNLOAD program	Table C—6	C—16
Error processing		
RELOAD program, file conversion	C.1.2.6	C—32
SCAN program	C.2.10	C—54
UNLOAD program, file conversion	C.1.1.7	C—15
Execution requirements, disc file UNLOAD program	C.1.1.6	C—14

F

Term	Reference	Page
Field specifications		
descriptor card type A	6.2	6—5
descriptor card type B	6.3	6—30
descriptor card type C	6.4	6—39
descriptor card type D	6.5	6—42
descriptor card type E	6.6	6—49
descriptor card type F	6.7	6—56
File conversion		
description	4.1	4—1
emulation aids	Appendix A	
requirements	4.2	4—2
File DVC, emulator load module		
diagnostics	6.7	6—59
how to specify	6.7	6—59
Form size		
diagnostics	6.6	6—50
how to specify	6.6	6—50
FORTRAN conversion		
compilation differences, OS/3 and 9200/9300	Table 4—1	4—4
OS/3 provided compilers	4.3.2	4—4

H

Term	Reference	Page
Hardware requirements		
emulated program	6.2	6—4
UNLOAD program	C.1.1.8	C—18
Header card, 8410 disc dump		
control stream sequence	Fig. A—2	A—4
format	Table A—1	A—5
function	A.1.1.1	A—4

I

Term	Reference	Page
Image mode option		
diagnostics	6.2	6—12
how to specify	6.2	6—12
Initial program load procedure, 9200/9300 emulator	B.1	B—1
Input requirements		
RELOAD program	C.1.2.1	C—18
SCAN conversion routine	C.2	C—40
UNLOAD program	C.1.1.3	C—4
	Table C—1	C—5
Interface requirements		
RELOAD program	C.1.2.4	C—25
SCAN conversion routine	C.2.8	C—54
UNLOAD program	C.1.1.2	C—3

J

Term	Reference	Page
JCL, EMULAT phase limitation		
diagnostics	6.7	6—64
how to specify	6.7	6—64
Job control interface, RELOAD program	C.1.2.4	C—25
Job control stream, UNLOAD program	Fig. C—5	C—15
Job execution, $Y$RUN library		
diagnostics	6.7	6—57
how to specify	6.7	6—57
Job priority		
diagnostics	6.7	6—62
how to specify	6.7	6—62
Job processing, logical approach to emulation	Fig. 1—1	1—4

L

Term	Reference	Page
Library name, emulator alternate load library	6.7	6—60
Line number, channel punching		
diagnostics	6.6	6—51
how to specify	6.6	6—51
Lines printed per inch		
diagnostics	6.6	6—53
how to specify	6.6	6—53
Load DVC, emulator loading from alternate disc pack		
diagnostics	6.7	6—59
how to specify	6.7	6—59

Term	Reference	Page
Load procedure, 9200/9300 emulator	B.2	B—2
Loading emulator from alternate disc pack		
diagnostics	6.7	6—59
how to specify	6.7	6—59
Logical unit number, 90/30 disc drive for system or data files		
diagnostics	6.3	6—30
how to specify	6.3	6—30
Logical unit number, 90/30 tape containing system or data files		
diagnostics	6.5	6—42
how to specify	6.5	6—42
M		
Main storage capacity, 9200/9300 system		
diagnostics	6.2	6—6
how to specify	6.2	6—6
Main storage requirements		
disc file RELOAD program	C.1.2.7	C—33
emulator	6.7	6—61
	7.2	7—1
Messages		
emulation	See commands/messages, emulation.	
emulatory advisory	B.4.3	B—15
RELOAD program	C.1.2.9	C—35
	Table C—8	C—36
	Table C—9	C—36
	Table C—10	C—40
Model specification for 0768 printer	6.2	6—20
Mount number, 9200/9300 disc pack		
diagnostics	6.3	6—31
how to specify	6.3	6—31
O		
Operating considerations		
descriptions	7.1	7—1
device assignments	7.6	7—2
disc prep	7.5	7—2
emulation, 8411/8414	7.4	7—2
emulator main storage requirements	7.2	7—1
instructions, 9200/9300 emulation	Appendix B	
minimizing program looping	7.7	7—2
priority assignments	7.7	7—2
running SG$EMJCL under spooling	7.3	7—1

Term	Reference	Page
Operating system identification, 9200/9300 system		
diagnostics	6.2	6—7
how to specify	6.2	6—7
Optical document reader, required by emulated program		
diagnostics	6.2	6—26
how to specify	6.2	6—26
Output files		
SCAN routine	C.2.7	C—49
UNLOAD program	C.1.1.4	C—8
P		
Pack-file label, 9200/9300 disc assignment	6.3	6—33
Paper tape reader required by emulated program		
diagnostics	6.2	6—26
how to specify	6.2	6—26
PARAM card preparation, RELOAD program	C.1.2.3	C—20
Peripherals		
conservation	7.6	7—2
emulation counterparts	Table 1—1	1—2
Physical unit number, 9200/9300 disc drive		
diagnostics	6.3	6—34
how to specify	6.3	6—34
Physical unit number, 9200/9300 tape drive		
diagnostics	6.5	6—43
how to specify	6.5	6—43
PIMAGE routine		
control card format	A.1.2.1	A—7
data restoration, dump tape to 90/30 disc	A.1.3.3	A—11
disc extent information processing	A.1.3.4	A—12
initialization	A.1.2.2	A—11
input tape verification	A.1.3.1	A—11
operational phases	A.1.3	A—11
parameter specification	A.1.2.1	A—9
prepping 90/30 disc input tape dump	A.1.3.2	A—11
restrictions and considerations	A.1.4	A—14
spooling procedure	A.1.4	A—14
Primary/secondary feed station substitutes, 1001 reader		
diagnostics	6.4	6—39
how to specify	6.4	6—39
Printer assign's for emulated program		
diagnostics	6.2	6—15
how to specify	6.2	6—15

Term	Reference	Page
Printer loop name		
diagnostics	6.6	6—49
how to specify	6.6	6—49
Program conversion		
BAL	See BAL conversion.	
COBOL	See COBOL conversion.	
FORTRAN	See FORTRAN conversion.	
RPG II	See RPG II conversion.	
Program directives, 9200/9300 emulation	B.4.1 B.4.4.1	B—4 B—15
Program loading, emulator	B.2	B—2

R

Term	Reference	Page
Recording mode characteristics, 9200/9300 tape		
diagnostics	6.5	6—44
how to specify	6.5	6—44
RELOAD program		
conversion mechanism	C.1.2.2	C—19
error processing	C.1.2.6	C—32
input files	C.1.2.1	C—18
job control interface	C.1.2.4	C—25
main storage requirements	C.1.2.7	C—33
messages	C.1.2.9 Table C—8	C—35 C—36
PARAM card format	Table C—9 Table C—10	C—36 C—40
PARAM card preparation	C.1.2.3	C—21
parameter specifications	C.1.2.3 C.1.2.3	C—20 C—21
Restoring 8410 disc images to 90/30 disc	A.1.2	A—7
RPG II conversion		
description	4.3.4	4—6
compiler differences between 9200/9300 and OS/3	4.3.4	4—6

S

Term	Reference	Page
SCAN conversion routine		
data base	C.2.9	C—54
error processing	C.2.10	C—54
function	C.2	C—41
input/output information specification	C.2.7.2	C—50
input requirements	C.2.6	C—47
module sequencing	C.2.7.1	C—50
output	C.2.7	C—49
related interfaces	C.2.8	C—54
related software components	C.2.8	C—54
SG$EMJCL, running in spool mode	7.3	7—1
Software environment, emulated program	6.2	6—4
Spooling, procedure for PIMAGE	A.1.4	A—14
Stub card option, emulated program		
diagnostics	6.2	6—14
how to specify	6.2	6—14
Supervisor/job control loading, system resident device		
diagnostics	6.2	6—9
how to specify	6.2	6—9
Switch priority, job/task control		
diagnostics	6.7	6—62
how to specify	6.7	6—62
SYSRES pack identification, 9200/9300 system		
diagnostics	6.3	6—35
how to specify	6.3	6—35
System console specification, 9200/9300 emulator	B.3	B—2
System hang due to emulator	7.7	7—2
System preparation, emulator generation	3.1	3—1
System resident device, supervisor loading		
diagnostics	6.2	6—9
how to specify	6.2	6—9

T

Term	Reference	Page
Tape drive identification, mounting SYSRES tape		
diagnostics	6.5	6—46
how to specify	6.5	6—46
Tape drives		
emulated program, defining	6.5	6—41
logical unit number specification	6.5	6—42
SYSRES tape unit identification	6.5	3—42
Tape loops, EMULAT phase limitation		
diagnostics	6.7	6—64
how to specify	6.7	6—64
Tape mode characteristics	Table 6—2	6—45
Transients, included in emulator generated		
diagnostics	6.2	6—8
how to specify	6.2	6—8

Term	Reference	Page
Type A descriptor card	6.2	6—4
Type B descriptor card	6.3	6—29
Type C descriptor card	6.4	6—38
Type D descriptor card	6.5	6—41
Type E descriptor card	6.6	6—48
Type F descriptor card	6.7	6—55

U

Term	Reference	Page
UNLOAD program		
error messages	Table C—6	C—16
error processing	C.1.1.7	C—15
execution	C.1.1.6	C—14
function	C.1	C—1
hardware requirements	C.1.1.8	C—18
input requirements	C.1.1.3	C—4
	Table C—1	C—5
interface requirements	C.1.1.2	C—3
job control stream	Fig. C—5	C—15
objectives	C.1.1	C—3

Term	Reference	Page
output files	C.1.1.4	C—8
preparation	C.1.1.5	C—12
structure	Fig. C—4	C—14
Utilities, conversion	See conversion utilities.	

V

Term	Reference	Page
Volume number, 90/30 disc pack (emulator alternate load library)		
diagnostics	6.7	6—60
how to specify	6.7	6—60
Volume serial number 90/30 emulator alternate load library pack		
diagnostics	6.7	6—60
how to specify	6.7	6—60
Volume serial number, 90/30 system or data files pack		
diagnostics	6.3	6—32
how to specify	6.3	6—32
VSN identification, 8411/8414 emulation	7.4	7—2

Comments concerning this manual may be made in the space provided below. Please fill in the requested information.

System: ____________________

Manual Title: ____________________

UP No: __________ Revision No: __________ Update: __________

Name of User: ____________________

Address of User: ____________________

Comments:

CUT

FOLD

FIRST CLASS
PERMIT NO. 21
BLUE BELL, PA.

BUSINESS REPLY MAIL NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SPERRY UNIVAC

P.O. BOX 500
BLUE BELL, PA.
19422

ATTN: SYSTEMS PUBLICATIONS DEPT.

FOLD

CUT