

SOFTWARE UPDATE
NOTICE
9200/9200 II
9300/9300 II

RELEASE MEMO # 147

DATE APRIL, 1975

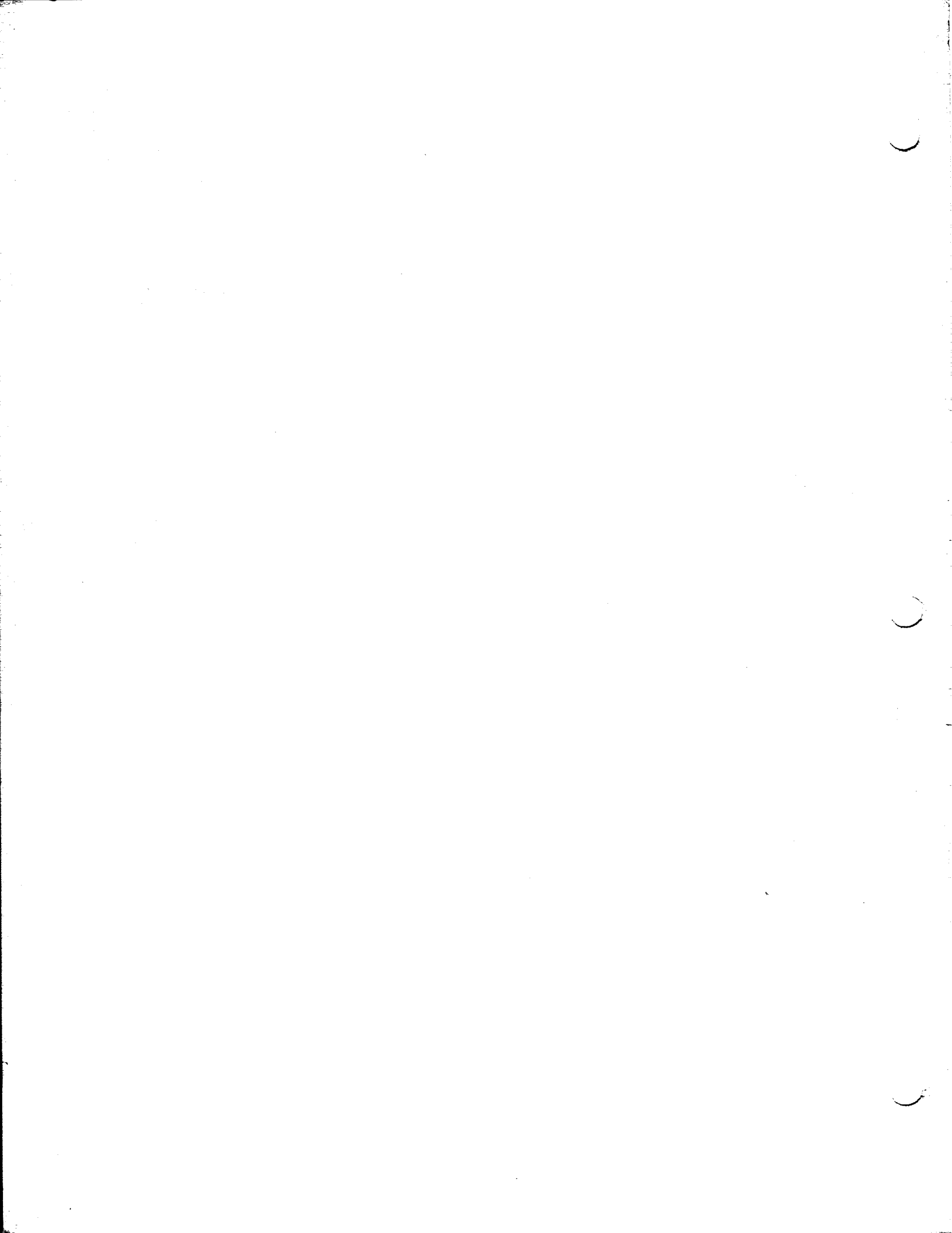
PRODUCT: 9200/9200II/9300/9300II
8411/8414 DNCOS REVISION 5.0.0

DISTRIBUTION

To Lists SA, RS, UA, UB, 174, 175, 176, 177, 178, 179, 184, 185, 186
187, 188, 189, 18A, 17A - Memo & Attachments

PHILADELPHIA DEVELOPMENT CENTER
SOFTWARE DEVELOPMENT SUPPORT
BLUE BELL, PA

SPERRY  **UNIVAC**
COMPUTER SYSTEMS



A. SOFTWARE UNIT:

<u>Ident.</u>	<u>Rev.</u>	<u>Block Count</u>
Master 8411/8414 System Preparation Tapes	5.0	
Master Tape #1		
Master Tape #2		

B. REASON FOR RELEASE:

1. Several enhancements to the RPG compiler, including the calculation time EXCEPTION operation.
2. Initial release of a tape cross-referencing utility and a disc cross-referencing utility.
3. Initial release of the 8411/8414 Direct Processor IOCS.
4. The capability of adding records to a sequential disc file.
5. Implementation of the System Execution Area and its related software.
6. Revisions to many areas of Disc NCOS software, including Sort Merge, Transients, Supervisor and Job Control, Assembler and Linker, Librarian, Dump/Restore and the Sequential Disc IOCS.

C. SOFTWARE RESTRICTIONS:

See individual Appendices for discussion of restrictions.

D. HARDWARE CONFIGURATION - Minimum for System Generation:

1. 24-K Memory
2. Two disc drives (8411 or 8414)
3. Card Reader (type Ø711)

E. SPECIAL NOTES:

1. Extensive changes have been made to all areas of the disc software. It is suggested you do not attempt to generate or use this software without first reviewing this release memo.
2. All loadable modules have been upgraded to the latest revision level. The loadable supervisor includes the Control Stream from Disc feature, which may be used for system generation. The loadable R11T supplied is the latest version and includes the "Source Code from Disc Feature".
3. Only one version of this disc software release has been prepared; its loadable supervisor uses the standard card reader (type Ø711) as a control stream reader. If it is necessary to generate a system which will employ a different control stream read device, the generation must be done at an installation which has the standard reader or else the loadables at the beginning of each master disc must be replaced with loadables generated and linked for the appropriate control stream reader before generating the final system.
4. The default System Job Control Stream file for the Job Control program supplied herewith is SYSFILE (SJCS=SYSFILE).

5. The loadable non-concurrent supervisor which is supplied for system generation purposes includes the following channel assignments.

<u>L.U.#(Hex)</u>	<u>Channel</u>	<u>Device</u>
	ALL HEX	
00-07	30-37	8411/8414 Disc Drives
08	01	Card Reader
09	02	Serial Punch
0A	03	Bar Printer
0B	07	1001
0C	06	Row Punch
0D	05	Optical Document Reader
0E	0C	Paper Tape
0F	0B	Drum Printer

6. The following parameters were used to generate the loadable Supervisor in this release. This supervisor occupies 4,992 (X'1380') bytes.

AFTR=ALL
 SIZE=24
 CSR=CRD
 DEVA=8
 SJCS=YES
 SYST=8411
 DCU1=X'30'
 UNT1=8
 DUMP=TEMP

7. There is a loadable Disc Report Program Generator on the master tape/disc with a base address of X'1380'. It was linked with 3 disc label parameter areas reserved. If this compiler meets the installation's requirements, the user can omit the linking of R11T, and use the loadable R11T included in this release.
8. There is no objection to users substituting other program names for UNIVAC-supplied utilities, e.g. "PRGM PHASEZRO,*" may be changed to "PRGM DSORT,*" or whatever other name is desired in the linker deck. Names of overlays (PHASE statements) must not be changed however, nor may names of programs within the RPG or COBOL compilers be altered. Two versions of an RPG compiler (R411 plus R11T, Rev. 3 plus Rev 4, etc.) must not be present in the same sysfile.
9. The following appendices are included in this release:
- A Master Tape/Disc Table of Contents
 - B System Job Control Stream
 - C System Generation Procedures
 - D Transient DNCOS 8411/8414 RPG
 - E 8411/8414 Sort Merge
 - F Implementation of DNCOS in 16-K Memory
 - G Tran and Transients
 - H Supervisor and Job Control
 - I 8411/8414 Disc Assembler (KSMB)
 - J 8411/8414 Disc Linker (DL11)

K Librarian (LIBS)
L Miscellaneous IOCS
M OS-500
N 8411/8414 Disc Utilities (DASM, DPRT, ISF)
O Dump/Restore Utility (DPRS)
P Tape Cross-Reference Utility
Q Disc Cross-Reference Utility
R NOTE Job Control Card
S SKIP Job Control Card
T VOL Job Control Card
U Sequential Disc IOCS (DSPIT AND DTFSD)
V Direct Processor
W DTFMT, DMTIO, DMTIT
X The System Execution Area
Y Halts

10. All SUR's that were to be corrected in Revision 5.0 are cleared with this release.

F. ORDERING PROCEDURES:

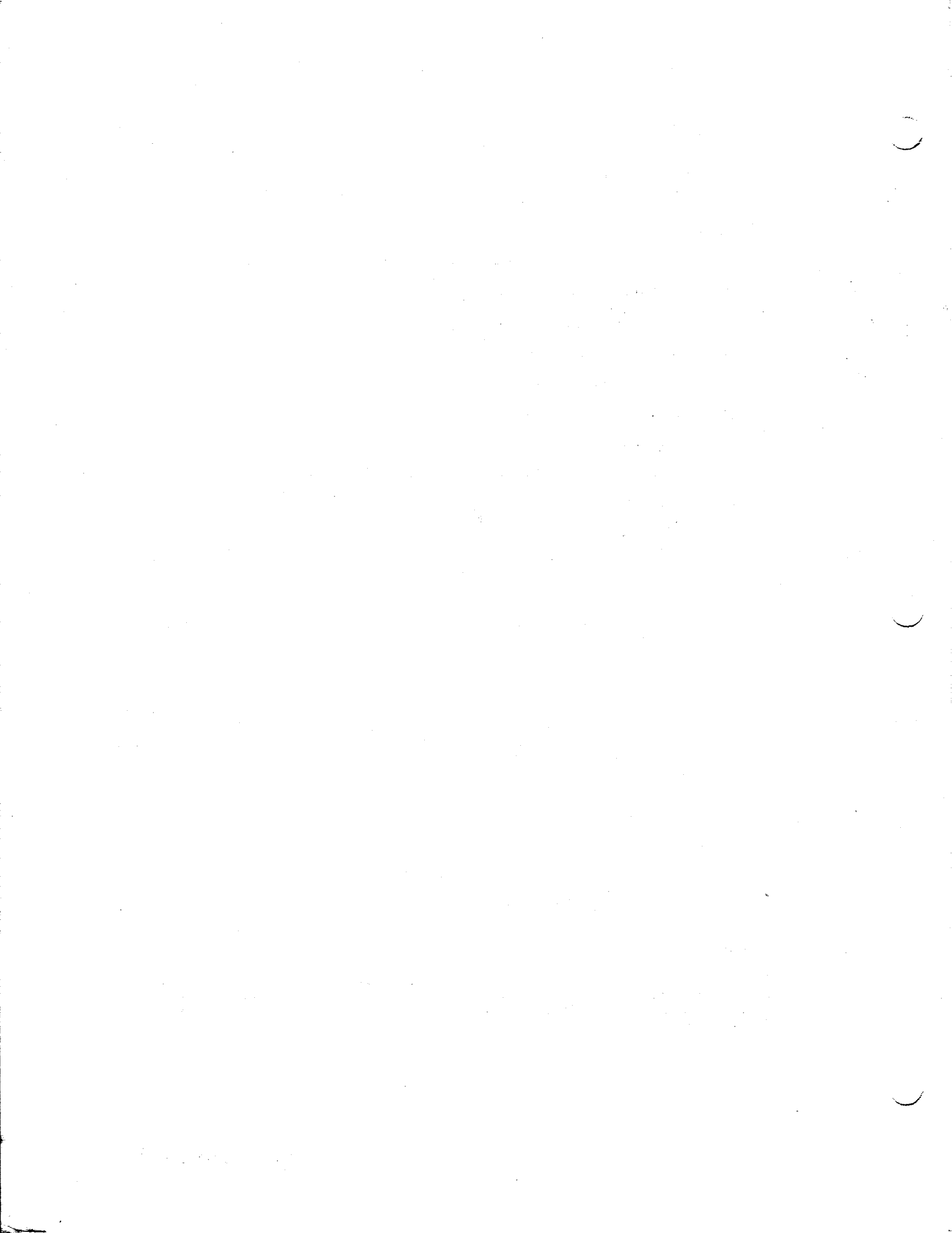
Master 8411/8414 System Preparation Tapes Rev. 5.0 are available with this release on tape or 8411/8414 Disc. The software may be ordered by forwarding the attached "Software System Support Request" directly to Software Order Services with applicable Software Media. i.e. two tapes or two 8411 disc or 1 8414 disc unit.

Sperry Univac 9200/9300 Software Coordinator
PDC Software Order Services
P.O. Box 500, Jolly & Township Line Roads
Blue Bell, Pennsylvania 19422

Note: All users outside of the United States should check with their local UNIVAC subsidiary office for distribution procedures. The above applies to United States Customers only.

G. DESTRUCTION:

This memo supersedes and orders the destruction of 9200II/9300/9300II Memo #141, dated August, 1974, and its updates. All corrections and changes described in Memo #141's update memos have been incorporated in this release.



APPENDIX A. MASTER TAPE/DISC TABLE OF CONTENTS.

Below are listed the contents of this software release. They are broken down into "Tape #1" and "Tape #2". The library group "TAPESYS" on Tape #1 is deleted from software which is shipped on disc. 8414 recipients receive the entire release in SYSFILE on a single disc pack. 8411 recipients receive the contents of Tape #1 in SYSFILE on one pack and of Tape #2 in SYSFILE on another pack. The loadable SUPR and JBCN at the beginning of Pack #1 are duplicated at the beginning of Pack #2 for 8411 recipients.

TAPE #1

BOB	TAPESYS		041775
ELT	L	TAPESYS	SUFF
ELT	L	TAPESYS	JBCN
ELT	L	TAPESYS	DIS
ELT	L	TAPESYS	DASH
ELT	L	TAPESYS	TDLIBS
ELT	L	TAPESYS	TDLIBSA
ELT	L	TAPESYS	TDLIBS4
ELT	S	TAPESYS	TPTC8411
ELT	S	TAPESYS	TPTC8414
ELT	L	TAPESYS	JBCN
ECC	TAPESYS		C41775
ELT	L		SUPR
ELT	L		JBCN
ELT	L		DL11
ELT	L		TRAN
ELT	L		KSMB
ELT	L		FHASEZRC
ELT	L		DASH
ELT	L		DFRT
ELT	L		ISF
ELT	L		DPRS
ELT	L		LIBS
ELT	L		LIBS4
ELT	L		LIBSA
ELT	L		XREF
ELT	L		SHEAF
ELT	L		R111
ELT	L		USERCEF
ELT	L		TPG1
ELT	L		USEFI
ELT	L		TPG2
ELT	L		LSEFC
ELT	L		TPG5
ELT	L		R110
ELT	L		IG25
ELT	L		CAL
ELT	L		IG43
ELT	L		USERSUBR
ELT	L		IG41
ELT	L		USERTABL
ELT	L		IG42
ELT	L		E
ELT	L		M
ELT	L		L
ELT	L		S
ELT	L		F
ELT	L		IG44
ELT	L		R30C
ELT	L		R20C


```

BOG      = AZ3
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZE
ELT S    FAZ3
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZB
ELT S    FAZE
BOG      MACROLIB
ECC      PACFOLIBUNIVAC
BOG      MACROLIBUNIVAC
ELT S    PACFOLIBUNIVAC
ELT S    MACROLIBUNIVAC
BOG      MACROLIBUNIVAC
ELT S    PACFOLIBUNIVAC
ECC      PACFOLIBUNIVAC
EOG      MACROLIBUNIVAC
ECC      MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ELT S    MACROLIBUNIVAC
ECC      PACROLIBUNIVAC
EOG      MACROLIBUNIVAC
ECC      PACROLIB
BOG      RLOCATBL
ELT R    FLOCATBL
ELT R    FLOCATBL
ELT R    FLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL
ELT R    RLOCATBL
ELT R    FLOCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL
ELT R    RLOCATBL
ELT R    RLOCATBL
ELT R    FLCCATBL

```

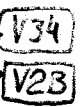
BOG TRANREL

```

041875
SYS1     5.0 SYSGEN #1
SYS2     5.0 SYSGEN #2
SYS3     5.0 SYSGEN #3
SYS4     5.0 SYSGEN #4
LTDS     R11T LINKER DECK
CDS      LINK UTILITIES
LMSD     SAMPLE DECK
LTU      LINK TAPE UTILITIES
SFR      SAMPLE FORMATTER RUN
CHANGE3  FOR CHANGING "SYS3"
041875
041475
041475
041575
SLPF
JBCN
041575
LUPC
#16
041575
041575
041475
DTFIS    VF
DTFDS    V6
DSPIT    VD
DAMIT    V1 2/26/75
DTFSD    V8
DTFDH    V1 2/26/75
DMTIT    V5
ISAM     VE 10/15/73
041475
041475
041475
041875
DL1      VD
DTS      VE
TBTF     VD
TMTP     V8
KA10     V8
KA11     V15
KA20     V8
KA21     V5
KA30     V8
KA40     V7
KA41     VE
FHSC     V8
PHSA     VE
PHS1     VE
PHS2     VE
PHS3     VE
PHS4     VE
PH4A     V6
PH5E     VE
S?QTTPUT VC
SAM?OPN1 VB
SAM?OPN2 VC
S?QCLOSE VB
S?QTGET  VB
IS?MCPEN VE
IS?CLOSE VE
IS?BADIN VE
IS?LADIN VE
IS?HIADD VE
IS?BSETL VE
IS?LSETL VE
IS?GETPT VE

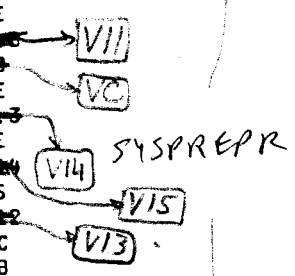
```

Test file
↑
↓



SYSPREP

MACEFILE



SYSPREP

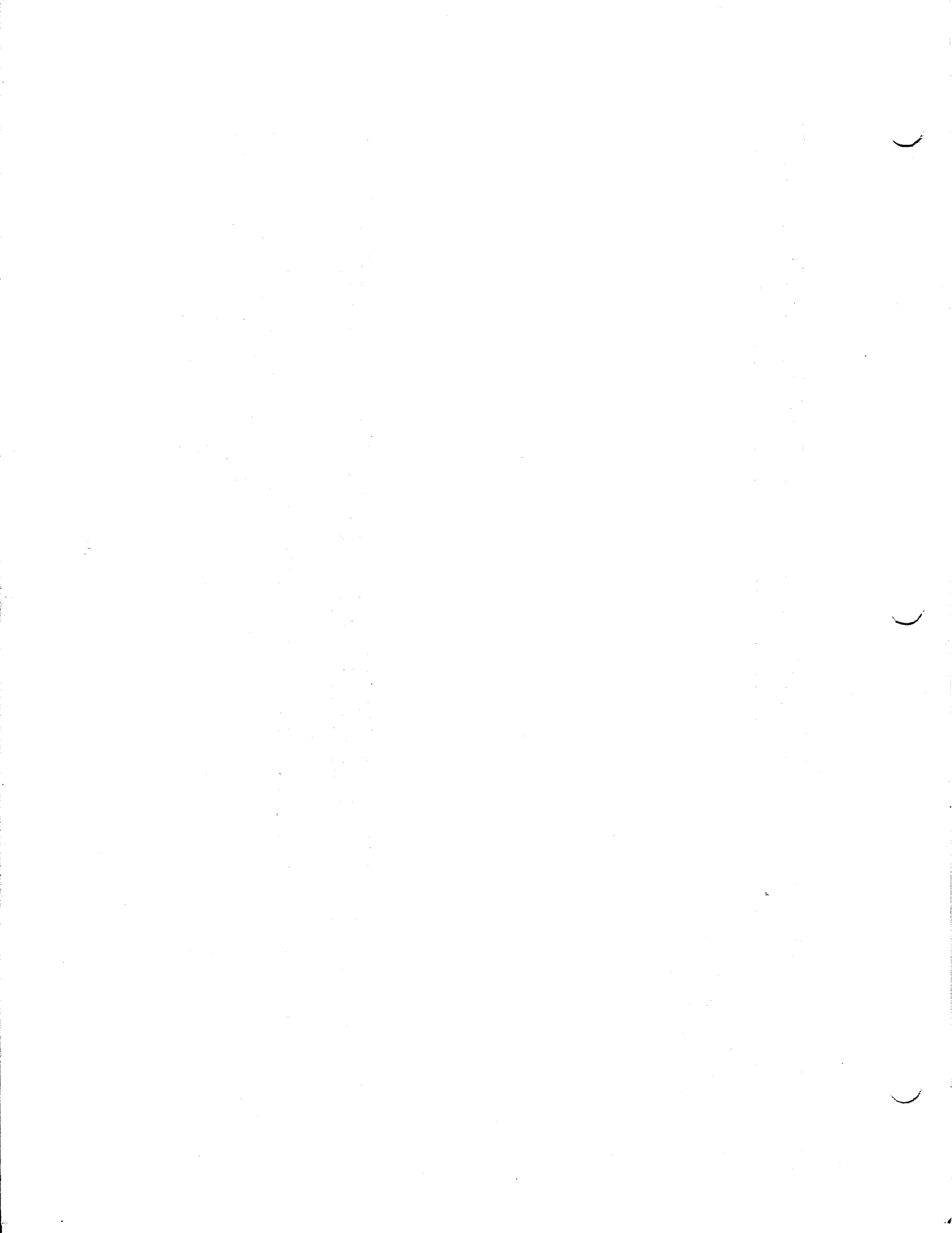
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLOCATBL
ELT R RLCCATBL
ECC FLOCATBL
END

ASEB J0
ETO A J2
G?XX J0
IBP8 J0
ETOX J1
XTOE J1
MPDP J3
EDIT J1
SHEAF1 J0
041875
041875

SYSPRCP2

+

Run



APPENDIX B. System Job Control Stream

A new feature, System Job Control Stream is now available to 8411/14 9200/9300 users.

SJCS will process control streams that have been filed on the system medium without precluding the use of the control stream card reader as a means of presenting control streams to the Supervisor and Job Control.

With the exceptions of the FINIS and SKIP Job Control Command Cards, images presented from either media will have the same function as they have had in the past. Further details on the function of the FINIS and SKIP Command Cards and a new Job Control Command Card RUN will be presented in subsequent paragraphs.

The current limitations on the use of SJCS are:

1. The system medium must be 8411/14, that is, SYST = 8411 (or 8414) in both SUPR and JBCN and,
2. The control stream card reader cannot be the 1001.

At initial program load (IPL) or

when a FINIS image from disc is read (no FINIS loop when read from disc) or

after a cancel (program, external or SKIP) the control stream card reader will process its control stream and will continue to do so until a RUN card is detected, at which time (if error free) the system (disc) job control stream will be processed until one of the three conditions mentioned above occurs (IPL, FINIS, Cancel).

One should be careful in placing DATE, DVCDN, DVCUP, JOB or SWAP images in the SJCS because of their pre-empting or being pre-empted by similar cards entered in the card reader control stream.

When processing the control stream from the system medium all demarcation records (X'55' in the first position) such as ELT, BOG, EOG: etc, except END, will not be presented to either Job Control or the user program but will be skipped over to the next sequential record. END demarcation records, when detected as the next image to be presented will cause a halt to occur and when run is depressed a cancel will take place.

A supervisor generated to process SJCS will be 680(10) bytes larger than the same supervisor generated without this capability.

The RUN Control card is used to direct Job Control to the System Job Control Stream to find its next control card statement. The System Job Control Stream is defined as being a control stream residing on the system device.

The format of the RUN control card is:

LABEL	△ OPERATION △	OPERAND
/	RUN	JOBNAME [, XX] [, FILENAME] B COMMENTS

■ Positional Parameter 1

JOBNAME - the name of the element containing the control stream to be presented to Job Control. The jobname is the name used in the ∇ ELT card when the element was placed on the system device.

■ Positional Parameter 2 (optional)

,XX - the logical unit containing the jobname to be performed. If present, it is specified in two hexadecimal digits.

omitted - logical unit $\emptyset\emptyset$ is assumed

■ Positional Parameter 3 (optional)

,FILENAME - the filename in which the jobname will be found. If present, it is specified by up to 8 alphanumeric characters.

omitted - the filename that was specified when Job Control was generated will be assumed.

EXAMPLES:

	LABEL	OPERATIONS		OPERAND
		8	16	
1.	/	RUN		DISCPRT
2.	/	RUN		ALPHA, $\emptyset 1$
3.	/	RUN		BETA, $\emptyset 1$, JOBFIL
4.	/	RUN		GAMMA, $\emptyset 2$, CNTRLST

	JOBNAME	LOG UNIT	FILENAME
EX 1 -	DISCPRT	$\emptyset\emptyset$	*
EX 2 -	ALPHA	$\emptyset 1$	*
EX 3 -	BETA	$\emptyset\emptyset$	JOBFIL
EX 4 -	GAMMA	$\emptyset 2$	CNTRLST

*the filename used will be that which the parameter SJCS was equated to when Job Control was generated.

SJCS Example

In the example that follows, (System Job Control Stream Example) List 1 represents a method of filing jobs into a file (SYSJCS) on logical unit 00. The LIBS being used to do this was linked with

```
DEL EQU X'55'      12-0-2 Control Punch
SENT EQU X'58'     2-8 (: ) Sentinel Punch
```

Note the use of the "VELT element name,*" command to allow the use of LIBS Command Cards and Job Control Command Cards in the elements being filed on the system medium.

Note also it is assumed the Job Control program was generated with SJCS=YES so that the filename used as a default in RUN command cards will be SYSJCS.

Presenting the control stream of RUN command cards as shown in List 2 will cause the following to occur: (any reference to XREF in command cards is for example purposes only)

- | | |
|----------|--|
| SXREF | 1. Element GENERATE located in SYSFILE on logical unit 02 will be listed |
| SXREF | 2. Element CDS with the same location will also be listed |
| SXREF | 3. Element JBCN with the same location will be cross-referenced |
| SXREF | 4. The FINIS will direct Job Control to the card reader for its next command |
| VTOCDASM | 5. A DASM will take place which will list the VTOC of the disc on drive 01 |
| VTOCDASM | 6. The FINIS will direct Job Control to the card reader for its next command |
| DSCPRINT | 7. A DPRT in hex-char mode of data from drive 00 will take place. The data from Cyl 119 head 00 will print followed by data from Cyl 119 head 02 through Cyl 119 head 03 followed by data from cylinder 200 head 00. |
| DSCPRINT | 8. Since this job does not end with a FINIS (and if a cancel does not occur) the next job filed will take place (VTOCDASM) |
| VTOCDASM | 9. Steps 5 and 6 will be repeated here |
| REMI | 10. A listing and cross-reference of an Element called REMI located in SYSFILE on logical unit 01 will take place |
| REMI | 11. At conclusion of the above, Job Control is directed to find the jobname VTOCDASM on logical unit 00 (default) in filename SYSJCS (default) and process it. |
| VTOCDASM | 12. Steps 5 and 6 will be repeated here |

- 0002 13. A halt display of 0002 demonstrates the successful conclusion of the control stream presented in List 2
- FINIS 14. If any programmed or external cancel had taken place in any of the above steps (or if a SKIP command card had been presented and made a match with the UPSI byte) then none of the subsequent RUN commands (or jobs) would have been executed and the control stream would have concluded with a FINIS loop.

List 1

List 2

<pre> / DATE '060574' / EXEC LIBS / DATA C ▽ CTL N,W,00, SYSJCS,R,00, SYSPFILE ▽ ELT REM1,* / EXEC XREF / DATA C ▽ CTL N,W,00, SCRATCH1,R,01, SYSPFILE ▽ FIND /REM1,,,,XREF ▽ INS /* / RUN VTOCDASM / /: ▽ ELT SXREF,* / EXEC XREF / DATA C ▽ CTL N,W,00, SCRATCH1,R,02, SYSPFILE ▽ FIND /GENERATE ▽ INS ▽ FIND /CDS ▽ INS ▽ SELECT MACROLIB/UNIVAC/EXECMODL/JBCN,,,XREF /* / FINIS / /: ▽ ELT DSCPRINT,* / EXEC DPRT / DATA T 00 02 011900 011900 00 02 011902 011903 00 02 020000 020000 /* / /: ▽ ELT VTOCDASM,* / EXEC DASM / DATA T DVC 01 END /* / FINIS / /* / FINIS </pre>	<pre> / RUN SXREF,00, SYSJCS / RUN VTOCDASM / RUN DSCPRINT / RUN REM1 / PAUSE 0002 / FINIS </pre>
	<pre> REM1 </pre>
	<pre> SXREF </pre>
	<pre> DSCPRINT </pre>
	<pre> VTOCDASM </pre>

SJCS Definitions

- Element - the record that contains the Element name is called a Demarcation Record. Also called a Module.
- Demarcation Record- any record containing a X'55' in the first position. It, in turn, is a Beginning of Group (BOG), End of Group (EOG), Element (ELT), etc.
- Job - a procedure to be done by the computer made up of one or more programs.
- Jobname - the name of a job to be done. Specified in a RUN card and is the name of an Element (Module) under which one or more programs will be executed.

If the execution of a given Job under a RUN procedure does not conclude with a FINIS and if no program causes a cancel to take place, then the next Jobname sequentially filed on the system medium will be processed.

SJCS Halts

Display	Meaning	Action
4F01	The control stream being processed from disc is about to go beyond the upper extent of the file	depressing Run will cause a Cancel to take place. No options are allowed
4F02	An error has occurred in the CCW chain attempting to bring in the next block of images from the control stream file	No response or a zero Key-in will cause a retry of the CCW chain. Any other response will cause a cancel
4F03	The current image that would have been supplied to either Job Control or the program being executed is an END demarcation record	depressing Run will cause a Cancel to take place. No options are allowed
4F04	The filename cannot be found on the logical unit specified	see Action to 4F02 display (search for filename and jobname)
4F05	The filename has been found, but the jobname cannot be found in that file	see Action to 4F02 display

If it is desirable to cause the input of card data while processing a control stream which resides on disc, it is accomplished by the use of a card with a slash in column 1, "DATA" in columns 10-13, and "READER" in columns 16-21 in the disc control stream at the point where a "DATA" statement would otherwise be appropriate. The detection of this card by the Supervisor (actually done in the PARAM? transient) causes the next control stream image delivered to be the next available image from the "real" control stream reader. For any program which is using a standard DTFCS, this should be a DATA statement with C, T or B in column 16. When the program executes an EOJ macro, the Supervisor resumes reading control stream images from the System Job Control Stream file on disc.

Since the presence or absence of the / DATA READER statement determines, at the time the program begins execution, the source of that program's control stream input, all control stream input to a given program must be from one medium or the other. The program is never aware of the medium from which its control stream is being read. This feature may be utilized by any program which reads control stream input without that program's being reassembled or recompiled.

The following example illustrates a standardized assemble-link-execute control stream on disc in which both the assembler and the linked user program read data from the "read" reader:

LIBS Run to Add Standard Assemble-Link-Execute Run to SYSJCS File:

```

/      EXEC  LIBS
/      DATA C
▽ CTL  N,U,Ø,SYSJCS,R,Ø,SYSFILE
▽ ELT  ASMBLEX,*
/      EXEC  KSMB
/      PARAM Ø1Ø1,1Ø,W,Ø,TESTFILE
/      PARAM Ø111,1Ø,1,WORKONE
/      PARAM Ø121,1Ø,1,WORKTWO
/      PARAM Ø131,1Ø,Ø,PROCLIB
/      DATA READER
/      EXEC  DL11
/      DATA C
/      CTL  N,W,Ø,OBJFILE,R,Ø,TESTFILE,S,1,WORKONE
/      PRGM  TESTPROG,*
/      LIBE  RLOCATBL,Ø,RELOCLIB
/      INCLUDE /TEST

```

```

/*
/      EXEC  TESTPROG,ØØ,OBJFILE
/      DATA READER
/      FINIS
/:
/*
/      FINIS

```

Control Stream to Assemble and Test a Program:

```

/      RUN  ASMBLEX          (SYSJCS ON ØØ ARE DEFAULTS IN THIS SYSTEM)
/      DATA T
TEST  START Ø
      USING *,Ø,1,2
      etc.
      etc.
      END
/*
/      DATA T
      .
      .   input data to program just assembled and linked
      .
/*
/      FINIS

```

Control Stream Analysis

In the lists and chart that follow an analysis is made of the sequence of events (conditions) and command cards that are valid and invalid.

The command card list is a list of the 22 allowable command cards. Each card is followed by two numbers; the first number is the column in List 1 of the chart in which you will find that card, the second is the column in List 2.

'Other Conditions' are events that affect the validity of following command cards, of these, only Input Data is in List 2 of the chart.

To use the chart, find the intersection of a given event or command card in List 1 and for the same (or different) condition or command card in List 2.

The decoding of the entry in the intersection is found immediately after the chart. The condition or command card in List 2 takes place as the next event after the condition or command card in List 1 for the purposes of utilizing this chart.

Command Cards

ALT - 3,4	DVCDN- 2,1	⑦ FINIS- 2,2	④ ⑤ RSTRT- 3,4
④ ASSGN- 1,3	DVCUP- 2,1	JOB - 1,1	RUN - 2,3
① CALL - 2,3	② ③ [END] - 2,6	④ MTC - 3,4	⑥ SKIP - {2,3
② ③ DATA - 5,5	EXEC - 3,3	NOTE - 2,3	{6,3
DATE - 2,3	④ FILE - 3,4	② PARAM- 4,5	SWAP - 1,3
		PAUSE- 2,3	UPSI - 2,3
			VOL - 2,3

Other Conditions

IPL 2,- Card Load 2,- Input Date 5,7 Cancel 6,-

Notes

1. Symbiont (COS, JCCN) only
2. Not processed by Job Control; by Supervisor (PARAM), or a card I/O routine such as DTFCS (DATA and [END]).
3. Only if data present in control stream
4. Only if tape input/output
5. Only if system is tape oriented
6. SKIP function is either Skip (2,3) or Cancel (6,3)
7. FINIS image in system medium causes a reversion to the card reader for control stream.

	LIST 1	ASSGN, JOB, SWAP	CALL, DATE, DVCDN, DVCUP, END, FINIS, NOTE, PAUSE, RUN, SKIP, UPSI, VOL, IPL, CARD LOAD	ALT, EXEC, FILE, MTC, RSTRT	PARAM	DATA, INPUT DATA	SKIP, CANCEL
LIST 2		1	2	3	4	5	6
DVCDN, DVCUP, JOB	1	E∅		2	1	3	4
FINIS	2			2	1	3	4
ASSGN, CALL, DATE, EXEC, NOTE, PAUSE, RUN, SKIP, SWAP, UPSI, VOL	3			2	1	3	∅
ALT, FILE, MTC, RSTRT	4	E∅	E∅		D∅	3	∅
DATA, PARAM	5	E∅	E∅			3	∅
[END]	6	E∅	E∅	2	D∅		∅
INPUT DATA	7	E ₁	E ₁	D ₁	D ₁		∅

Code Meaning

blank Valid sequence

∅ ignored, card not processed - not an error

1 valid if loaded program does not require Input Data, 41FD halt display if Input Date required

2 41∅2 halt display if not following the EXEC card. Ignored if loaded program does not require Input Data and does follow EXEC. 41FD if following EXEC and Input Data required

3 processed as [END]

4 cards are processed and JOB and FINIS stop the Cancel procedure. DVCDN and DVCUP do not stop the cancel procedure.

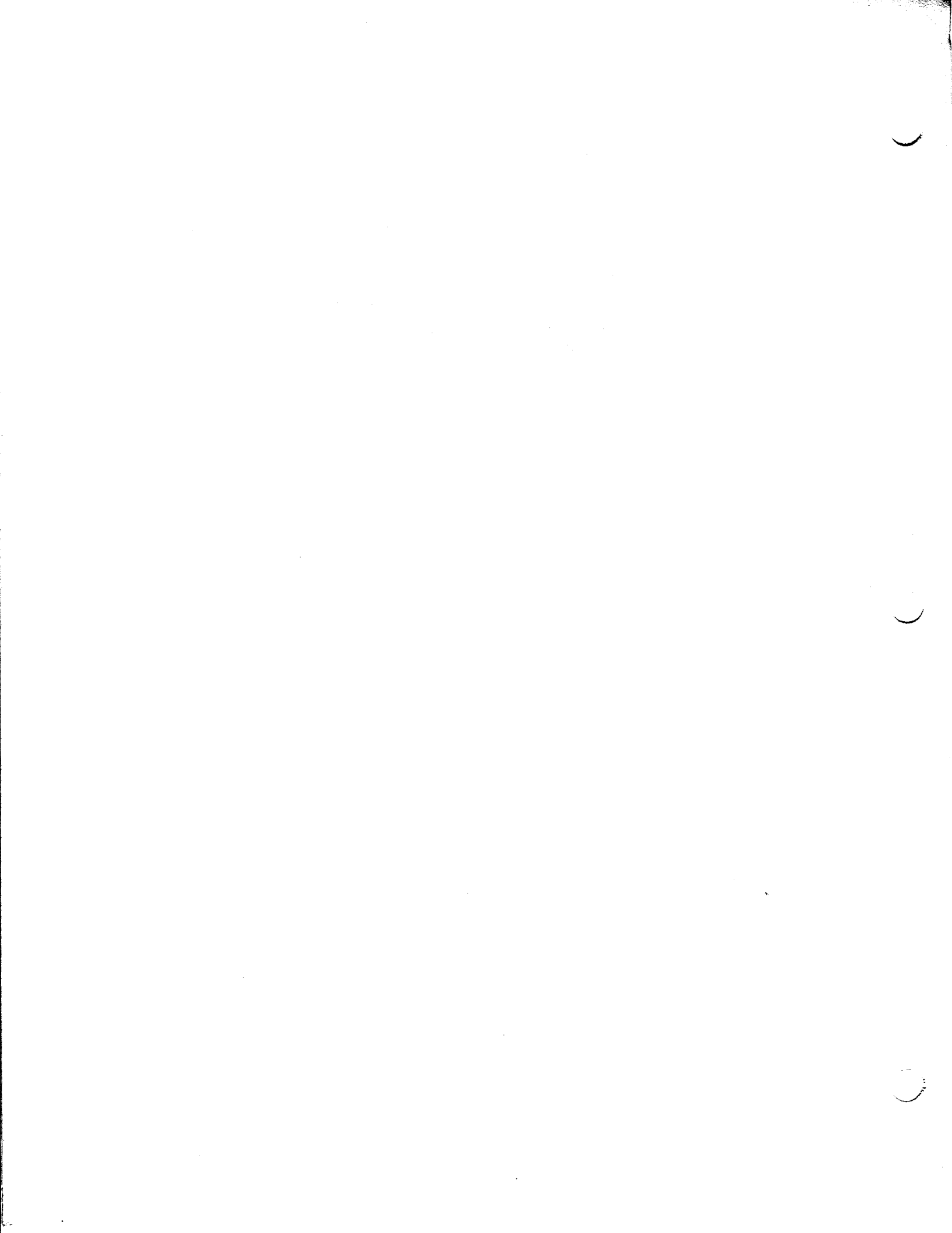
D0 4102 halt display if loaded program does not require Input Data (after executing loaded program). 41FD if Input Data required by loaded program.

D1 41CF halt display if following EXEC and loaded program does not require Input Data (after executing loaded program). Immediate 41CF if not following EXEC. 41FE if loaded program required Input Data.

E0 4102 halt display on detection of List 2 card

E1 41CF halt display on detection of Input Data

NOTE: Although a JOB card following an UPSI card is not an 'error' the JOB card will reset the UPSI byte to binary zero.



APPENDIX C. System Generation Procedures.

This Appendix contains a description of the steps to be taken to prepare a user's system disc using the master tape/disc received from UNIVAC. Those procedures which are described in 9200/9300 Series Operating System (UP-7531) will not be reiterated here except to point out recent changes or to clarify matters which have been the subject of questions. Most of the job control decks which are necessary for the accomplishment of systems generation are supplied on the master tape/disc. Each is catalogued in standard 9200/9300 Tape/Disc Librarian format and may be punched out with the appropriate SELECT or FIND statement. The FIND statement is recommended, as thus facilitating any alterations which might be necessary. It is also recommended that all decks punched by LIBS be interpreted for ease of handling. Each deck will be punched with an ELT card preceding it which must be discarded prior to placing the deck in the reader for execution. Where the character "▽" appears below, it is to be taken as representing the Librarian's control character (normally a 12-Ø-2 multipunch).

The loadable Supervisor and Job Control which are distributed on the master disc have the capability of running a control stream from disc (System Job Control Stream feature). If the control stream decks which are provided herewith fulfill the needs of a given installation, they may be run as is from disc by the RUN statement. They may also be updated, as desired, by LIBS prior to running so as to effect any necessary changes.

1. Prep all disc packs required for generation.

If new disc packs, or packs of uncertain quality, are to be used for System Generation, the appropriate version of the 8411/8414 Prep Program should be run to initialize those packs. (8411 users should prep a minimum of 3 packs, 8414 users 2). The Prep is supplied in card-loadable form to all new users; UP-7835 should be consulted for operating instructions. During an initial full Prep no options should be selected, i.e., make the following replies to the indicated halt displays:

<u>Display</u>	<u>Reply</u>
ØØ1Ø	FF
ØØ2Ø	FF
ØØ3Ø	Hardware device address of disc unit containing pack to be prepped, i.e., 3Ø, 31, 32, etc.
ØØ4Ø	ØØ

2. Copy master tapes to disc packs.

Those who receive this DNCOS release on disc packs should skip to paragraph 3; the procedures described in this paragraph are identical to those used by UNIVAC to copy its master tapes to disc pack(s) for shipment.

A small additional library group, TAPESYS, is included at the beginning of the first of the two master tapes. The purpose of this group is to enable the copying of the remaining contents of both master tapes to two 8411 disc packs or to one 8414 disc pack. TAPESYS includes the following elements:

SUPR	Tape-resident NCOS Supervisor
JBCN	Tape-oriented Job Control Program
DTS	Data Tape Services
DASM	Direct Access Space Management
TDLIBS	Tape/Disc Librarian (Serial Punch)
TDLIBSA	Tape/Disc Librarian (Row Punch)
TBLIBS4	Tape/Disc Librarian (1004 Punch)
TPT08411	Run stream for copying master tapes to 2 8411 packs
TPT08414	Run stream for copying master tapes to one 8414 pack
JBCN	Tape-oriented Job Control Program

The Supervisor in TAPESYS has tape drives on channel 8 assigned to logical units 0-3 and disc drives with controller base address X'30' assigned to logical units 4-7. Other logical unit assignments are identical with the disc-resident Supervisor described in the cover letter of this memo. The tape-resident supervisor was generated without any DUMP or REP capability.

To begin system generation, place master tape #1 on tape unit 0, and a scratch tape on unit 1. Punch the following run deck manually:

```

/      EXEC  TDLIBS      (or TDLIBSA or TDLIBS4)
/      DATA  C
      CTL  N,W,1,TAPE,R,0,TAPE
▽ FIND  /TPT08411,,,PUNCH  (or TPT08414 for 8414 users)
▽ INS
/*      REWIND
/      FINIS

```

A

Place the above run deck in the reader and boot the tape-resident system from X'C0'. A deck will be punched out by the Librarian which may be used to allocate disc file space and to copy the master tapes to disc (s). This run deck will also be listed on the bar printer. If the user wishes to delete undesired elements from the master disc or add elements of his own, the appropriate Librarian commands may be inserted in this run deck.

- 2a Dismount the scratch tape and mount master tape #2 on tape unit 1. Place the run deck punched by LIBS in the reader, mount disc pack (s) as necessary, and re-boot from X'C0' or restart the control stream by an X'E0' operator request. If X'1111' halts occur, X'E5' replies should be made. LIBS will transfer the contents of the master tapes to disc and provide a complete listing of all elements copied. When X'1234' is displayed, without feeding any cards in the reader, set X'30' in the data entry switches, then re-IPL from disc. The disc system will write out the transient file (TRANSCYL) and it is now possible to carry out the remainder of the System Generation from disc.

NOTE: Although a version of LIBS is provided which will punch on the 1004 punch, the tape/disc Supervisor has no 1004 capabilities and any punching which has to be done on a 1004 punch must be done under the control of a user-supplied supervisor.

PHASE I

3. Generate and/or Link SUPR, JBCN, DL11 and TRAN.

The procedures described in this paragraph are commonly referred to as "Phase I" of System Generation. Paragraphs 4 and 5 describe "Phase II", which consists of linking the remainder of the software including the RPG Compiler. Phase I is begun with master disc #1 on disc unit 0 and a scratch pack on unit 1. Material with brackets [] indicates coding or actions whose inclusion is optional.

There are four run decks provided with this release for the purpose of accomplishing Phase I of System Generation. The decks, all of which are in Sysfile on master disc #1, are as follows:

<u>Elt-name</u>	<u>Description</u>
SYS1	24-K NCOS system, no tapes, no SJCS feature; Supervisor size < X' [redacted] '.
SYS2	32-K NCOS system with tapes and SJCS feature; Supervisor size < X'1800'.

1080

<u>Elt-name</u>	<u>Description</u>
SYS3	32-K NCOS (OS-500) system with inquiry capability from the system console, roll-in/roll-out, SJCS feature, and no tapes; Supervisor size < X'2300'.
SYS4	32-K COS (OS-500) system with local and remote capabilities, single inquiry option, SJCS feature, print-buffering to logical unit zero, roll-in/roll-out, and magnetic tapes; Supervisor size < X'4000'.

The four Phase I decks just described were intended to serve only as samples of common DNCOS configurations. It is the responsibility of each installation to determine what system should be generated to meet its own needs. For this purpose, one or more of the UNIVAC-provided generation decks should be listed in order to ascertain what changes must be made to tailor the sample configuration to your own requirements. If desired, the generation deck may also be punched. To list [and punch] a Phase I generation deck, the following run deck should be punched manually:

```

/      EXEC LIBS          (or LIBSA or LIBS4)
/      DATA C
/      CTL N,W,Ø,SCRATCH1,R,Ø,SYSFILE
▽ FIND /SYSn[,,,PUNCH]
▽ INS
/*
/      FINIS

```

B

Place the above run deck in the reader and boot the UNIVAC-supplied loadable Supervisor from X'30'. A deck will be listed [and punched] by the Librarian which may be used as the basis of Phase I of System Generation. It executes the following:

SYSn

```

DASM  To allocate space on Unit One for Assemblies.
KSMB  To assemble the Supervisor.
KSMB  To assemble the Job Control Program.
KSMB  To assemble the Logical Unit/Physical Unit Tables.
KSMB  To assemble Printer/Punch Logical Units (PLUN).
[KSMB To assemble disc-to-print symbiont - SYS4 only.]
DASM  To re-allocate space on Unit One for the Linker.
DL11  To link the Supervisor.
PAUSE ØØØ1
DL11  To link Job Control, Linker, TRAN, disc-to-print symbiont .
*LIBS To punch out run stream decks for Phase II. (CDS & LTDS)

```

The Assembler source decks which are included in SYS1-SYS4 are identical with decks used at UNIVAC to generate systems for testing of DNCOS Rev. 5.0. Each keyword parameter for Supervisor generation is annotated. The listing of the entire deck should be compared with the generation instructions contained in UP-7531 and an evaluation made of the user's requirements. Unnecessary parameters may be deleted, additional options added, and other changes made to reflect the needs of a particular configuration. The PRGM statements for linking JBCN, DL11, and TRAN will need to be modified after linking the Supervisor to reflect the value assigned to the symbol "E?HI" during that linkage. For this reason, a PAUSE statement is included between the two executions of DL11.

*If this deck is to be run on a system with Row or 1004 punch, the last EXEC operand should be changed to LIBSA or LIBS4. Although a version of LIBS is provided which will punch on the 1004 punch, the disc-resident Supervisor has no 1004 capabilities and any punching which has to be done on a 1004 punch must be done under the control of a user-supplied supervisor.

3a If one of the 4 supplied Phase I decks meets your requirements as is, Phase I of System Generation may be run by placing the following 2 cards in the reader and re-booting from X'30' or restarting the control stream with an X'E0' operator request:

```

/      RUN   SYSn
/      FINIS

```

If it is necessary to modify one of the supplied run decks before accomplishing Phase I, the punched deck may be updated by manually inserting and deleting cards or the Librarian may be used to effect the change on disc so that the corrected generation deck may be run directly from disc. To assist in performing such an update, a small library file named SYSJCS is available which has a capacity of 223 card images on 8411 (849 on 8414). A sample control stream, CHANGE3, is provided which illustrates the modification of the SYS3 run to cause it to generate a non-inquiry OS-500 system which logs only JOB, EXEC, CALL and FINIS control stream images. CHANGE3 is listed herewith:

```

/      EXEC  LIBS
/      DATA C
▽ CTL  N,W,0,SYSJCS,R,0,SYSFILE
▽ FIND /SYS3
▽ COR  26
▽ COR  30,31
▽ COR  45
          CLOG=JECF
▽ COR  102,103
▽ COR  120
          PRGM  JBCN,X'1C00'
▽ COR  124
          PRGM  DL11,X'1C00'
▽ COR  129
          PRGM  TRAN,X'1C00'
▽ INS
/*
/      RUN   SYS3,00,SYSJCS
/      FINIS

```

This run would be executed by submitting the following cards in the control stream:

```

/      RUN   CHANGE3
/      FINIS

```

After making the necessary changes to a SYSn run deck by LIBS, the updated run could be executed by submitting the following control stream:

```

/      RUN   SYSn,00,SYSJCS
/      FINIS

```

OR a manually-corrected run deck could be placed in the reader and UNIVAC-supplied loadable Supervisor may be booted from X'30' or the control stream restarted by an X'E0' operator request. Upon successful completion of the run,

- SYSFILE on unit 1 will contain loadable Supervisor, Job Control, Linker, and TRAN, which are generated for the user's configuration.
- Run decks for the accomplishment of Phase II will have been listed [and punched out].
- A bootstrap block will be written on logical unit 1.

PHASE II

4. Link Utilities.

It is assumed that the remainder of the generation procedures will be carried out with the Supervisor whose generation was described in "3." above. If it is not possible to execute this Supervisor on the machine where generation is being done, consult Paragraph 7 of this Appendix for deviations involved in using the Supervisor on the master disc. If the Supervisor generated during Phase I includes the SJCS option the execution of Phase II may be accomplished by means of RUN statements to execute the appropriate combinations of CDS, LTU, and LTDS. If Phase II of System Generation is done through the RUN statement, the following cards must be inserted in the card reader ahead of the first RUN statement:

```
/      EXEC   TRAN,Ø1
/      DATA   T
/      (blank except for ØØ in cols. 39,40)
/*
```

This is necessary because Job Control's RUN routine makes use of transients.

CDS

The run deck labelled "CDS" may now be placed in the reader and the newly-generated Supervisor booted from X'31'. SJCS users may run it via:

```
/      RUN   CDS          NOTE:  LIBS linker cards supplied supports only
/      FINIS              disc.  Disc/Tape LIBS linking instructions
                          reference UP-8028.
```

-- TRAN will be executed to write transients on TRANSCYL on unit 1 for use during Phase II.

-- Then DL11 will be executed to link all UNIVAC-supplied utilities at the base address which will be in effect with the user's Supervisor. If the configuration includes magnetic tape units, the linking of DTS and TPRR may be accomplished by running the deck labelled "LTU" immediately after CDS. The sequence of linker statements for each utility should be maintained. To minimize possible errors, all linker statements are sequence numbered.

-- SHEAF is then executed to pre-format the System Execution Area and to write Job Control, LIBS and the Sort/Merge in it. Additional UNIVAC-supplied or user-supplied programs may be added to SYSEXAR at this point if desired. See Appendix X.

Upon completion of this run, SYSFILE on unit 1 will now contain loadable versions of all UNIVAC-supplied utilities. 8411 users must now dismount master disc #1 from unit Ø and mount master disc #2.

LTDS

5. Link RPG Compiler.

The run deck labelled "LTDS" may now be placed in the reader and the user's system booted from X'31' or the control stream restarted by an X'EO' operator request. The R11T compiler will be linked into a scratch file on Unit Ø, then copied by LIBS behind the other loadable elements in SYSFILE on Unit 1. This procedure is necessary due to the internal design of the compiler. Before carrying out this run, the RPG compiler link deck may be modified to reflect label areas for load-and-go compiler capability or non-standard printer logical unit number. (See Appendix D).

SJCS users may accomplish the linking of the RPG compiler via:

```
/      RUN   LTDS
/      FINIS
```

LMSD

6. COPY Relocatables and Macro Libraries.

SYSFILE on unit 1 now contains all loadable elements of UNIVAC software which will be required for the operation of the user's 9200/9300 system. If programming will be done in the Assembler language, such additional elements as are required may be copied from master disc #1 and/or #2. The Table of Contents contained in Appendix A should be consulted to determine the locations of the desired elements. Remember that all macros must be placed in "library groups" within "library files" to enable them to be located by the Assembler. Those macros whose use is unnecessary in the current transient DNCOS but whose use might be required under MOS or an earlier level of DNCOS which did not use transients are distributed on 8411 master disc #2 (or last part of 8414 SYSFILE) in library group, NONTRAN. The run deck labelled "LMSD" is a sample of an execution of LIBS to add macros and relocatables to SYSFILE.

7. Procedures for System Generation if Final Configuration Unavailable.

It is sometimes necessary to generate an operating system and other software on a machine other than the one on which the system will be used. This will normally present few problems if the procedures involved in systems generation are understood. The instructions contained in this appendix are intended only as general guidance. The specific procedures carried out will depend on the configuration differences which are involved.

If the Supervisor which is generated during Phase I cannot be operated during Phase II due to differing channel assignments, unavailable hardware, etc., it is suggested that the Supervisor provided on the master disc be used while linking the user's utilities and compilers. Instead of rebooting from X'31', as described in paragraph 4 above, the linking of utilities would be accomplished with the Supervisor which was already booted from X'30'. The execution of TRAN which occurs at the beginning of the "CDS" link deck would have to be deferred until the generated system could be operated on its final configuration. Since there are also copies of SUPR, JBCN and the transient file on 8411 Master Disc #2, the linking of the RPG compiler would proceed as described in paragraph 5 above with the Supervisor from X'30' being used instead of the Supervisor from X'31'. The use of the Linker on Logical Unit One insures that all elements are linked at the base address which will be in effect when the system is placed into operation.

If the user's linker is linked at a lower base address than can be used with the UNIVAC-supplied SUPR, as could happen when generating a 16-K system, the user's generated SUPR could be loaded from X'31' on a larger machine and used to run Phase Two as long as its memory size was changed to reflect the size of the machine being used. This is accomplished by keying in to the location represented by the tag "E?TM" a X'5F' or X'7F' so as to be able to take advantage of the larger machine's memory size.

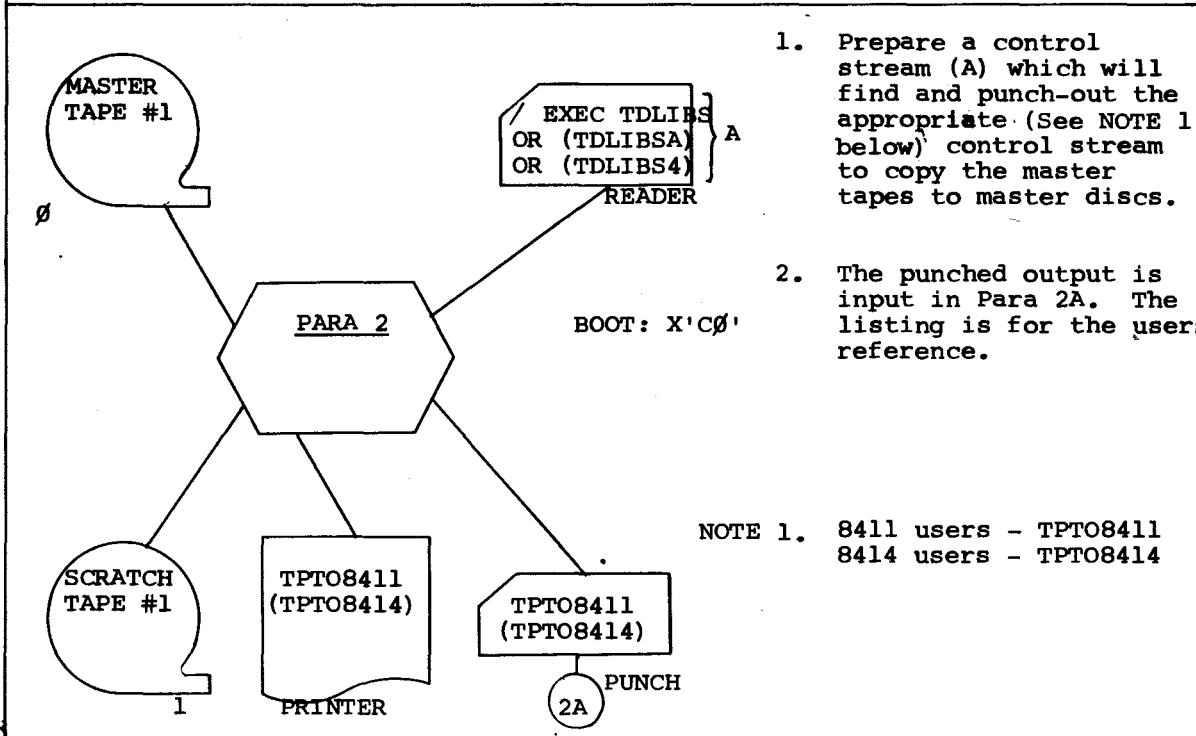
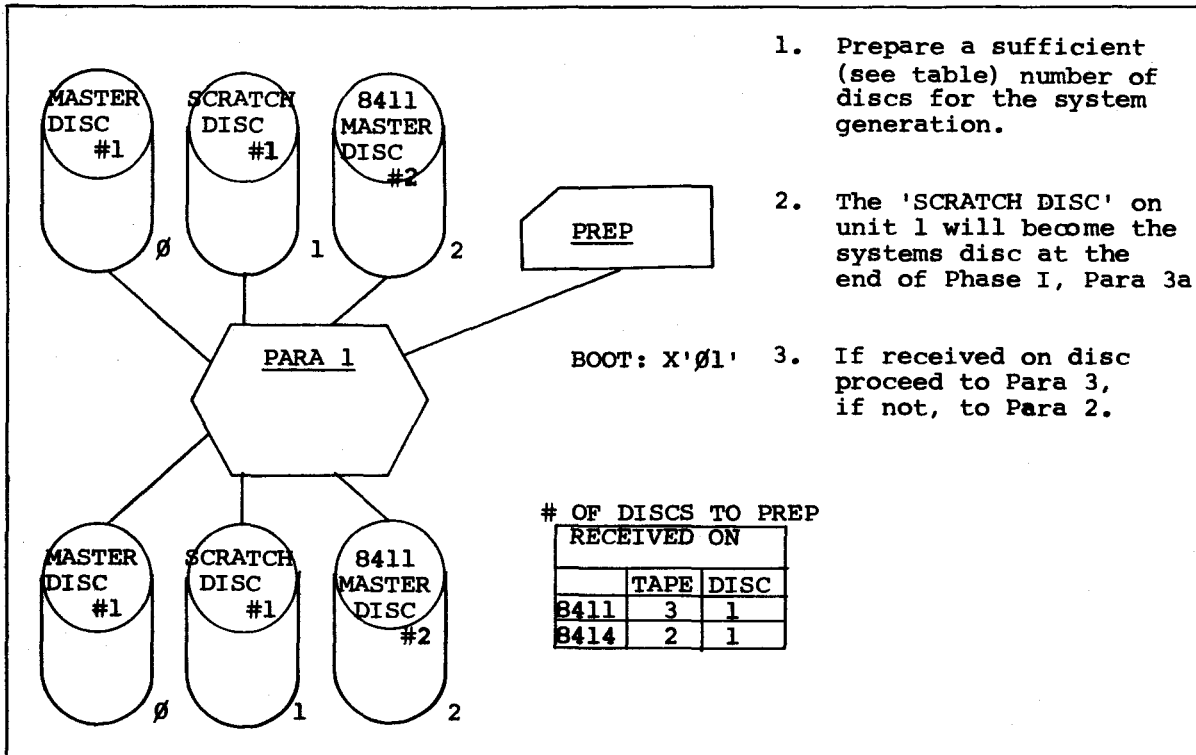
8. Concurrent Operating System Generation.

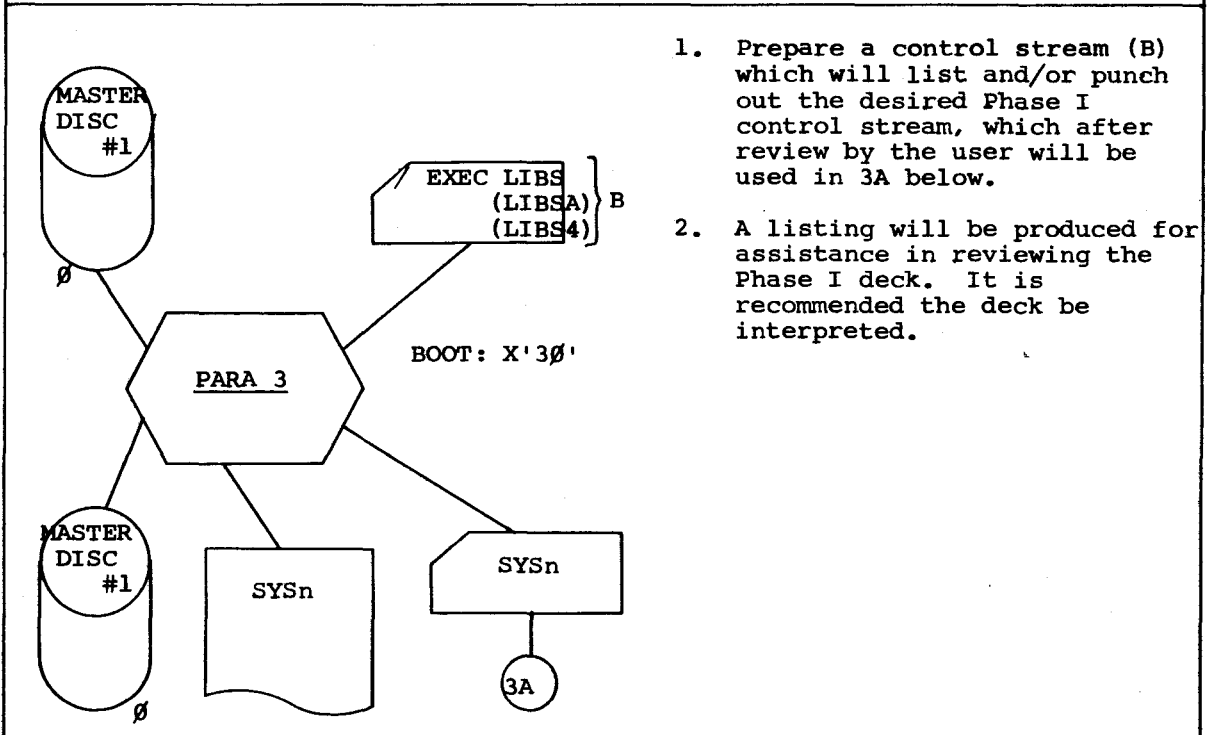
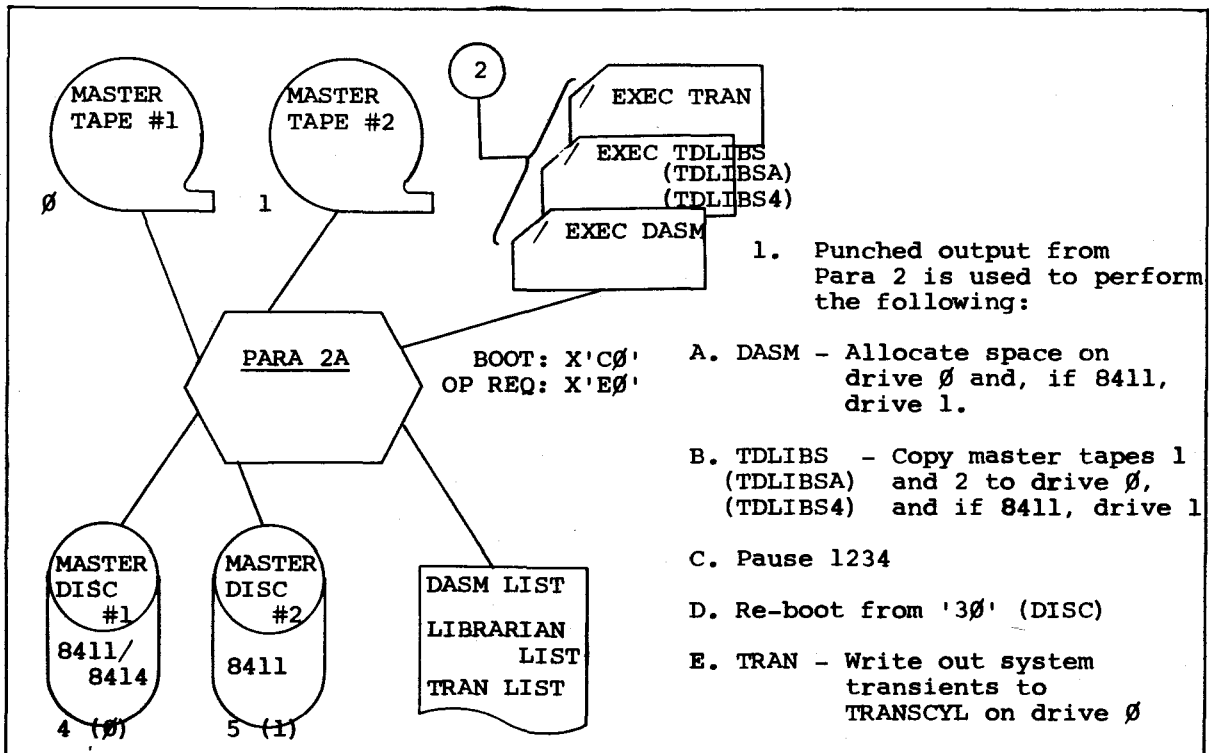
There are no significant differences to be observed in generating a concurrent operating system (COS). UP-7531 Rev. 5 describes possible changes to the system generation deck and linking of alternate print modules if buffered printing will be used.

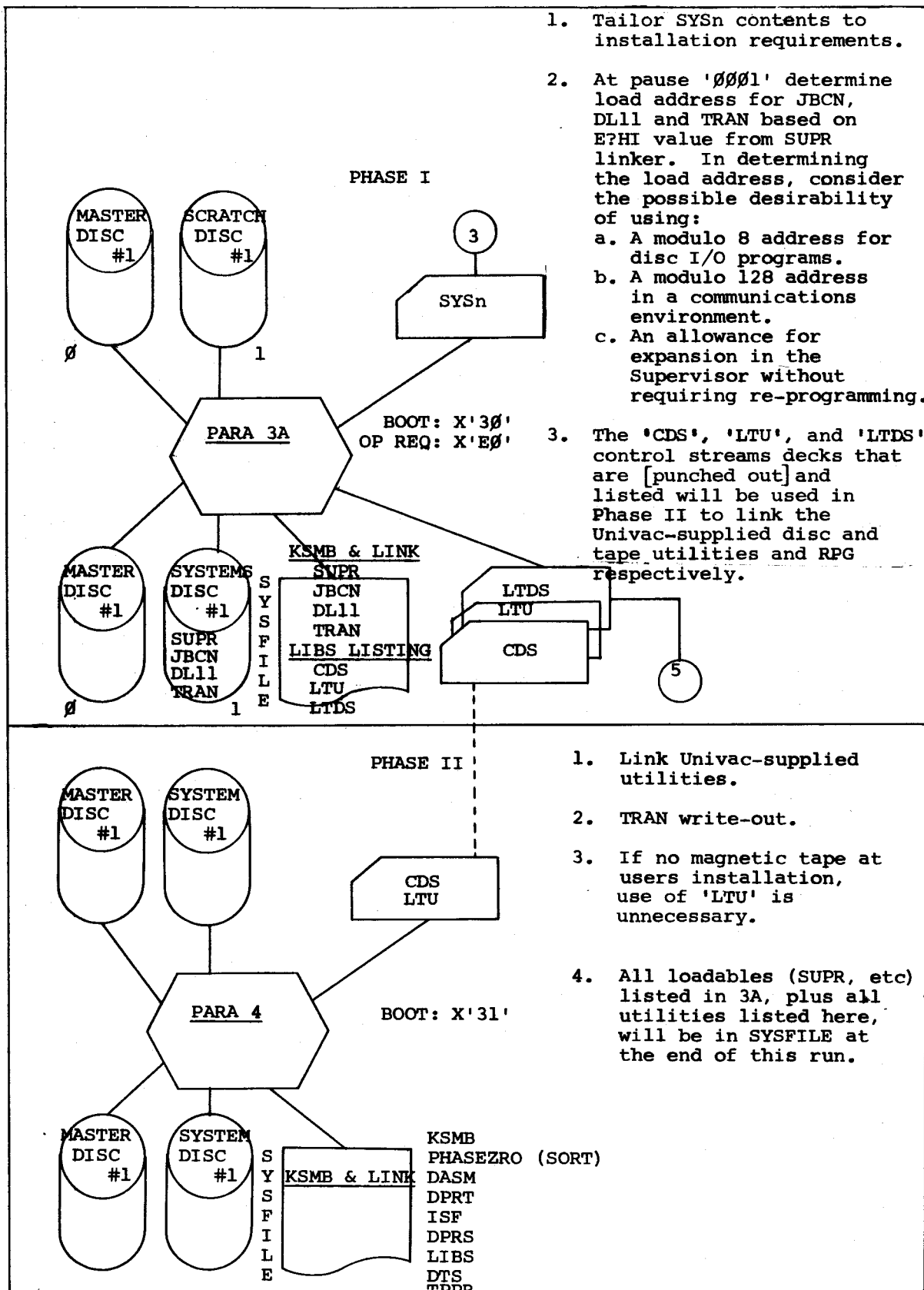
All print modules described in UP-7531 were assembled with DEVA=E. All were assembled with DTFPZ instead of DTFPR; if buffered printing is desired, DTFPR must be used. The element-names of print modules supplied with this release all have "TR" appended, e.g., DSP2TR is the element-name for the print module which was assembled with DSP2 as its label on its START line.

9. In generating the Supervisor and Job Control, there is no difference between specifying SYST=8411 and SYST=8414. Identical code is generated. Any operating system, UNIVAC software, (except the PREP), or user program which uses standard Disc IOCS which executes in an 8411 environment will execute with 8414's and vice versa, and will execute in a mixed 8411/8414 system. Installations which are contemplating the conversion from one type of drive to the other must only ensure that their Supervisor is generated with sufficient disc 8414 drives to an 8411 control unit; it IS possible, however, to connect 8411 drives to an 8414 control unit. This may be the simplest method of making the transition from 8411's to 8414's.

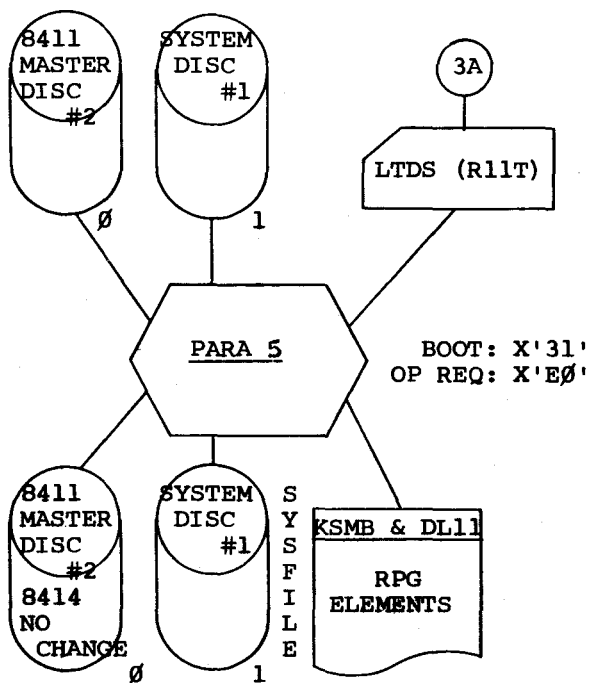
NOTE: Different volumes of a single file cannot be on both 8411's and 8414's.





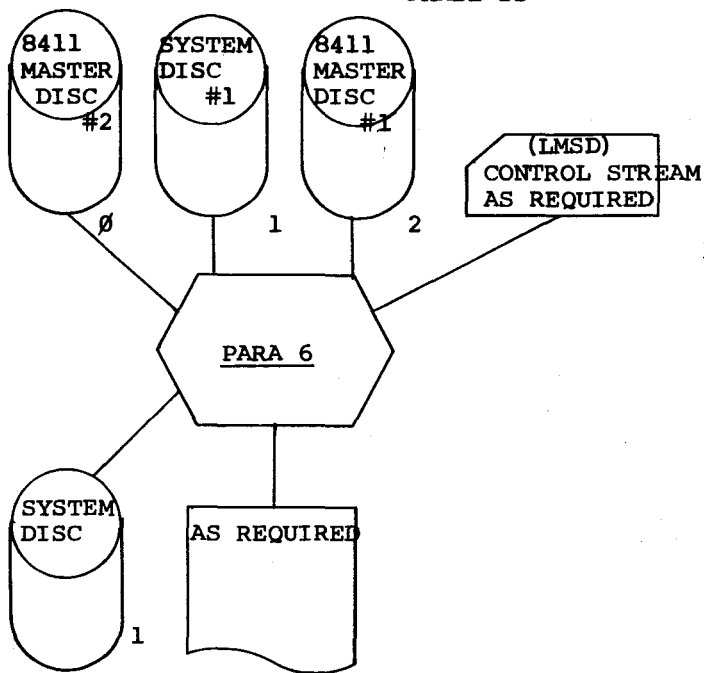


PHASE II (Cont.)

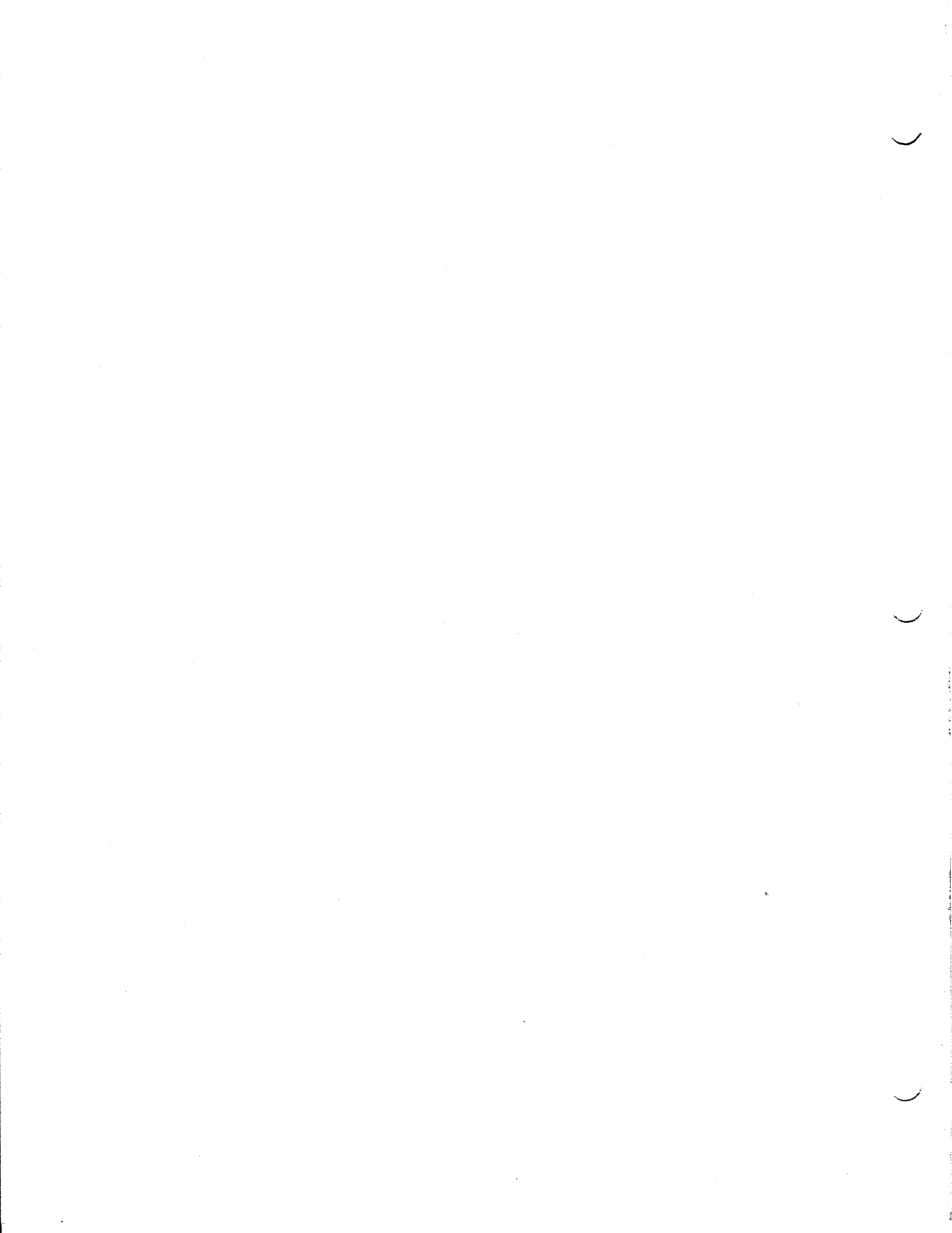


1. Modify the 'LTDS' control stream deck as necessary to obtain necessary label areas for 'LOAD-AND-GO' capability, or non-standard standard printer logical unit numbers (See Appendix D).
2. The 'LTDS' control stream punched out in 3A will be used to link the R11T compiler into a scratch file on unit 0. It will then be copied by LIBS into Sysfile (behind the other loadables) on Unit 1.

PHASE II



1. Create a control stream as required by the installation to copy whatever loadable, relocatable or source elements will be needed on the 'SYSTEMS PACK' for the efficient operation of the installation. LMSD may be used for this purpose.



APPENDIX D. TRANSIENT DNCOS 8411/8414 RPG

A. SOFTWARE UNIT:

<u>Ident.</u>	<u>Rev.</u>
R11T. Transient DNCOS 8411/8414 RPG	5.0

B. REASON FOR RELEASE:

1. To correct the following problems:

- a. The RGØ84I error diagnostic (failure to edit a numeric field on output) has been changed from an ØD14 halt to a non-fatal ØDØ1 halt.
- b. The out of sequence ØE11 message is changed to await a reply for OS-5ØØ systems.
- c. When OR indicators were specified with ADD output specifications, a 25ul display occurred.
- d. A Processor Abnormal condition occurred when a LIMITS tape was closed.
- e. The 8411/8414 SAM IOCS routines used to handle object time tables were updated to the most current version. The volume number check error that occurred is now eliminated.

2. To implement the following:

- a. The calculation operation EXCPT is now available in the R11T compiler. The Exception operation allows records to be outputted to different types of output files during detail or total calculation. (See Special Note 1)
- b. The End of Diagnostics (ØDØ1) halt and the End of Generation (Ø333) halt may now be bypassed. (See Special Note 2)
- c. R11T users may now force an end-of-job condition by turning on the Last Record (LR) indicator during calculation (detail/total) time. (See Special Note 3)
- d. R11T now has the capability of adding records to a sequential disc file. Adding is accomplished by writing new records starting at the address currently occupied by the end of file mark. (See Special Note 4)
- e. The number of halt indicators permitted has been expanded from two to eight. (See Special Note 5)
- f. When processing a control stream which resides on disc, it is now possible to allow RPG specification cards to be inputted from the serial card reader as well as from the disc file containing the control stream. (See Special Note 9 for information concerning this new feature).

C. SOFTWARE RESTRICTIONS:

1. When running a control stream from disc (SJCS option) that includes an RPG compilation, and the RPG source statements are embedded in the control stream, a processor abnormal stop will occur.
2. Secondary input files must be specified before chained files.
3. Unblocked Index Sequential Files must have embedded keys, i.e., there must be an entry in Columns 35-38 of the "F" card.
4. Standard labels must be used on a tape containing tables to be loaded at object time.
5. If the memory configuration is 24K, R11T cannot be linked at an address higher than $3B00_{16}$. If symbionts reside in memory with RPG, then this address must be reduced by the size of the symbiont(s).

D. SPECIAL NOTES:

1. EXCEPTION OPERATION

- a. This operation permits object-program output to one or more device types at any time during detail or total calculations. The function can be used in applications where:

1. A number of identical records are to be transferred to an output device in one RPG cycle by repeatedly specifying EXCPT.
2. A number of similar records are to be transferred to an output device in one RPG cycle. Those fields which are different may be modified for each output. They may also require calculation before the next record is placed on the output file. This can now be accomplished without having to repeat the specifications for the entire record. The exception lines function is of great advantage where a table with mathematically related entries is to be built. The object program can print, punch or write the table, one line at a time, within one RPG cycle.
3. Repetitive calculations chaining and updating or adding of records of a file is required before the cycle for one input record has been completed.

- b. Execution of an EXCPT operation causes each record, designated with an E in column 15 and conditioned by indicators (if any), to be placed into its associated file. The order in which these records are described on the output-format specifications form determines the sequence in which the object program places them into their output files.

c. Overflow Indicators

When an overflow condition is encountered, during the execution of exception output lines, the program sets the appropriate overflow indicator before it returns control to the calculation routines. Hence, either 'of' and/or 'OV' overflow indicators may be on when the program continues executing the remaining calculations.

d. Entries on the Calculation Specification Form

For an exception line operation, the specifications on the calculation specifications form are as follows:

<u>Column</u>	<u>Specification</u>
7-8	Control level indicator (optional)
9-17	Any valid conditioning indicator
28-32	The operation code EXCPT

See Example 1.

e. Entries on the Output-Format Specification Form

The only specification affected is the use of the letter E in column 15. This associates an output record with detail or total calculation time rather than detail or total output time. The specifications for all other columns remain the same.

f. Error Display ØEEE

When the EXCPT operation is being performed, if both the EXCPT calculation specification and at least one output specification with an E in column 15 are not present, the 'ØEEE' error display will occur. Correct the program and restart.

g. Restrictions

1. 7-track tape files without data conversion (TAPE7) cannot be used as output files in an EXCPT operation. This restriction will be removed in a future release.
2. Exception output records must be specified after total lines on the output-format specifications form. See Example 2.
3. An overflow indicator cannot condition an exception-output line.

UNIVAC

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM		FORM TYPE	CONTROL LEVEL	INDICATORS		FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	SELECTING INDICATORS	COMMENTS											
PG. NO.	LINE NO.			A	B								H	L	E										
3	5	6	7	11	12	14	15	17	18	27	28	32	33	42	43	48	49	51	52	53	54	55	58	60	74
	0.1	C				KEY1	CHAIN	FISAM					30	30											
	0.2	C	N30				EXCPT																		ISAM UPDATE
	0.3	C	30				EXCPT																		ISAM ADD
	0.4	C																							
	0.5	C																							

EXAMPLE 1

SPERRY UNIVAC

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM		FORM TYPE	TYPE N/E/T	STACKER SELECT	SKIP	PRINTRY OPTION	OUTPUT INDICATORS	ZERO SUPPRESS	BLANK AFTER	DATA FORMAT	CONSTANT OR EDIT WORD																
PG. NO.	LINE NO.			A	B	C	A	D																			
3	5	6	7	14	15	16	17	18	19	20	21	22	23	25	26	28	29	31	32	37	38	39	40	43	44	45	70
	0.1	O	PRINT	H			207				IP																
	0.2	O		OR							OF																
	0.3	O																					44		'HEADING'		
	0.4	O	PRINT	D			1																				
	0.5	O									KEY1																
	0.6	O									RECORD																
	0.7	O	PRINT	T			2				LR																
	0.8	O									SUM																
	0.9	O																									
	1.0	O	FILEONE	E							N30																
	1.1	O									RECORD																
	1.2	O																									
	1.3	O	FILEONE	E	ADD						30																
	1.4	O									KEY1																
	1.5	O									RECORD																

EXAMPLE 2

2. ØDØ1 and Ø333 Displays

This version of the Report Program Generator now has the ability to by-pass the end of diagnostic (ØDØ1) and the end of generation (Ø333) displays.

This feature is controlled by column 9 of the RPG header card. The codes for column 9 are as follows:

- blank - both X'ØDØ1' and X'Ø333' will be by-passed. If the X'Ø333' message is not displayed, then execution of a program must be from the object module.
- D - only the end of diagnostic (X'ØDØ1') message will be displayed.
- E - only the end of generation (X'Ø333') message will be displayed.
- B - both the end of diagnostic (X'ØDØ1') and the end of generation (X'Ø333') messages will be displayed.

3. Last Record (LR) Indicator

The LR indicator can now be SETON and SETOF during calculation (either detail or total) time. This feature permits the following:

- a. A program can be forced into an end-of-job condition. If a program is prematurely terminated and data remains in the control stream, then a X'41CF' will be displayed.

Note: All files will have been closed prior to X'41CF'
- b. When an end-of-job condition occurs, the LR indicator can be turned off during TOTAL CALCULATION time allowing the LØ loop (detail time) to be executed. It is then the responsibility of the user to return the job to normal termination by turning the LR indicator on again.

In order to further clarify the changes to the Last Record (LR) indicator, a revised RPG cycle has been enclosed in this release. (See Figure 2-8)

4. RllT now has the capability of adding records to an 8411/8414 sequential disc file. Adding consists of the ability to start writing new records at the address currently occupied by the end of file mark rather than being forced to start an output file at the beginning. Two rules must be followed to add to a SAM file:

- a. An "A" must be specified in column 66 (File Addition) of the File Description card pertaining to the SAM file.
- b. Output specification cards with the "ADD" designation in columns 16-18 should not be used for SAM files.

5. The number of halt indicators permitted in an RPG program has been increased from two to eight (H1 through H8). The halt displays $\emptyset E\emptyset F$, $\emptyset E\emptyset \emptyset$ and $\emptyset E\emptyset F\emptyset$ are no longer valid for R11T as of this release. H1-H8 halt indicators will now be displayed as follows:

H1 = $\emptyset B8\emptyset$
 H2 = $\emptyset B4\emptyset$
 H3 = $\emptyset B2\emptyset$
 H4 = $\emptyset B1\emptyset$
 H5 = $\emptyset B\emptyset 8$
 H6 = $\emptyset B\emptyset 4$
 H7 = $\emptyset B\emptyset 2$
 H8 = $\emptyset B\emptyset 1$

Each bit position in the right-most byte of the display represents a different halt indicator.

BIT POSITION:	\emptyset	1	2	3	4	5	6	7
HALT INDICATOR:	1	2	3	4	5	6	7	8

If more than one halt indicator is set at one time, the bits for the appropriate halt indicators will be OR'd together in the display. Therefore, for example, if halt indicators 1, 4, 7 and 8 are set, the display will be $\emptyset B93$.

6. Linking

IMPORTANT: The R11T linker deck is now set up for a 'GENERATE and GO' RPG with 3 disc PARAM areas, rather than with \emptyset PARAM areas as in the past.

- a. The R11T linker deck, called LTDS is punched out as the last step of the system generation run. If it is desired to punch this link deck separately from system generation, the following Librarian run may be utilized:

```

/      EXEC  LIBS      (or LIBSA or LIBS4)
/      DATA  C
@ FIND  /LTDS,,,PUNCH
@INS
/*      REWIND
/      FINIS

```

- b. The linker statements for linking R11T are provided in the RPG linker deck.
- c. The RPG linker deck uses the asterisk convention. Thus, R11T will be linked at the same location as the disc linker is linked.
- d. The linker deck is set up for a 'GENERATE and GO' RPG with 3 PARAM areas. If 3 PARAM areas are not desired, then changes must be made to the linked deck (refer to Special Note #8).
7. Procedure for changing the 'GENERATE and GO' Report Program Generator.

The purpose of a 'GENERATE and GO' RPG is to allow the storing of disc or tape label information via PARAM cards prior to the generation of an object program. A parameter area of 26 bytes is required for each tape label (standard labels only; non-standard labels are not permitted) and 58 bytes for each disc label. It should be noted that this will decrease the memory remaining for the generation which in turn decreases the memory size for the object program correspondingly. This is wasteful for any object program which does not require the full number of PARAM areas.

To increase or decrease the size of the label areas, the following changes must be made to the R11T linker deck prior to Phase 2 of the system disc generation.

- a. The PRGM which

	COL. 10	16
--	---------	----

specifies	PRGM	programname,*+936+174
should be changed to	PRGM	programname,*+936+L

where L is the number of labels times 26 and/or 58

- b. The EQU card which

	COL. 1	10	16
--	--------	----	----

specifies	L	EQU	174
should be changed to	L	EQU	n

where n is the number of labels times 26 and/or 58

- c. Card #'s R11T0080 and R11T0090 (INCLUDE /PRM1 and PRM2) are used to reserve the required number of bytes necessary for storing PARAM information. These modules reserve an area of 174 bytes (58 bytes times 3). The following six modules are provided for use in conjunction with the PARAM areas:

TAPE:

LEN1 (reserves 26 bytes for 1 PARAM)
 LEN2 (reserves 52 bytes for 2 PARAM's)
 LEN4 (reserves 104 bytes for 4 PARAM's)

DISC:

PRM1 (reserves 58 bytes for 1 PARAM)
 PRM2 (reserves 116 bytes for 2 PARAM's)
 PRM4 (reserves 232 bytes for 4 PARAM's)

By specifying different combinations of these modules, any number of PARAM areas (up to the permissible number of devices) may be reserved.

EXAMPLE

In the following example an installation requires 2 disc labels and 1 tape label for a 'GENERATE and GO' environment.

- a. The PRGM card which

	COL. 10	16
--	---------	----

specifies	PRGM	TPG1,*+936+174
should be changed to	PRGM	TPG1,*+936+142

- b. The EQU card which

	COL. 1	10	16
--	--------	----	----

specifies	L	EQU	174
should be changed to	L	EQU	142

d. Linker Deck Changes

The following is a list of the specific cards to be changed for either a zero PARAM Report Program Generator or a 'GENERATE and GO' Report Program Generator which requires less than or more than 3 PARAM areas.

PRGM cards

R11T0200, 0280, 0490, 0900, 1630, 2850, 3140, 3280, 3500,
3630, 3750, 3850, 3940, 4030, 4130, 4240, 4340, 4510,
4610, 5030, 5190, 5900

EQU cards

R11T0030, 0210, 0290, 0500, 0910, 1640, 2860, 3150, 3290,
3510, 3640, 3760, 3860, 3950, 4040, 4140, 4250, 4350,
4520, 4620, 5040, 5200, 5910

8. For additional information regarding PARAM areas, refer to UP-7620, Rev. 2.
9. Reading RPG specification cards from the control stream card reader when processing a control stream which resides on disc.

It is now possible to allow RPG source programs to be inputted from either the control stream card reader or from a disc file when processing a control stream which resides on disc. This is accomplished by placing a card with a slash in column 1, "DATA" in columns 10-13, and "READER" in columns 16-21 in the disc control stream at the point where a normal "DATA" statement would be located. The detection of this card by the Supervisor causes the next control stream image delivered to be the next image from the "real" control stream reader. For more information concerning the "DATA READER" option, see Appendix B, System Job Control Stream, page 6.

A sample control stream to compile and execute an RPG program follows:

LIBS Run to Create a Control Stream on Disc:

```
/ EXEC LIBS
/ DATA C
∇ CTL N,W,Ø2,CNTLFILE
∇ ELT COMPILE,*
/ EXEC R11T
/ PARAM (DISC OR TAPE - the number of PARAM's allowed
is governed by the size of the compiler's
PARAM area)
/ DATA READER
/ EXEC RUN1,Ø1,RPGOUTPT
/ DATA READER
/ FINIS
/:
/*
/ FINIS
```

Control Stream to Compile the RPG Programs:

```
/  RUN    COMPILE,Ø2,DNTFILE
/  DATA  C
      (RPG Source Cards to compile an object pgogram
      named RUN1)
/*
/*
/  DATA  C
      .
      .  input data to program just compiled
      .
/*
/  FINIS
```

NOTE:

When compiling more than one RPG source program in succession via the control stream reader, the normal job control card sequence between the source programs is:

```
/*
/*
/  DATA  C
```

If the DATA C card is inadvertently omitted, miskeypunched, etc., a processor abnormal will occur. This condition will be corrected in a future release.

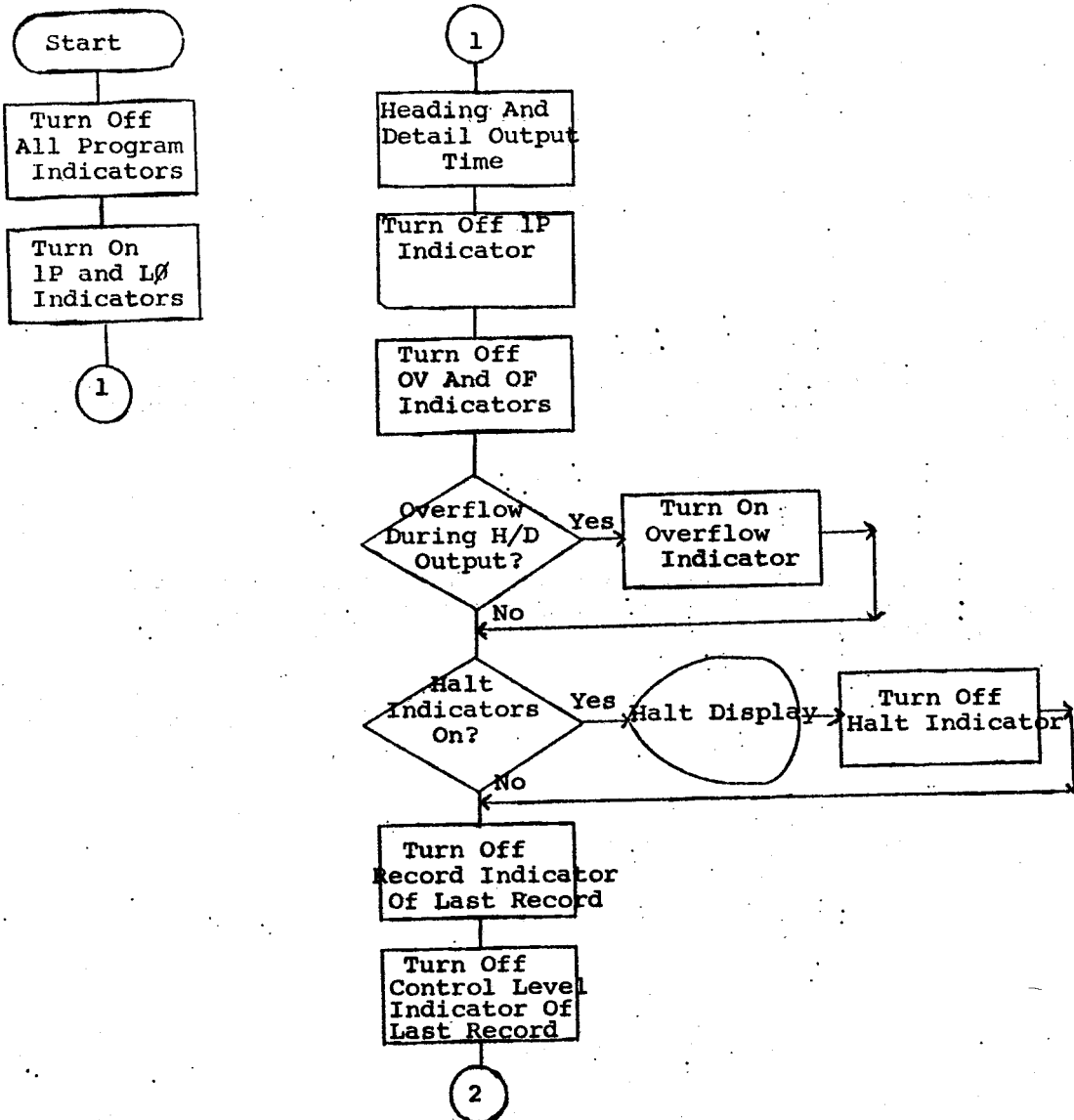


Figure 2-8. Operating Cycle Flowchart (Part 1 of 3)

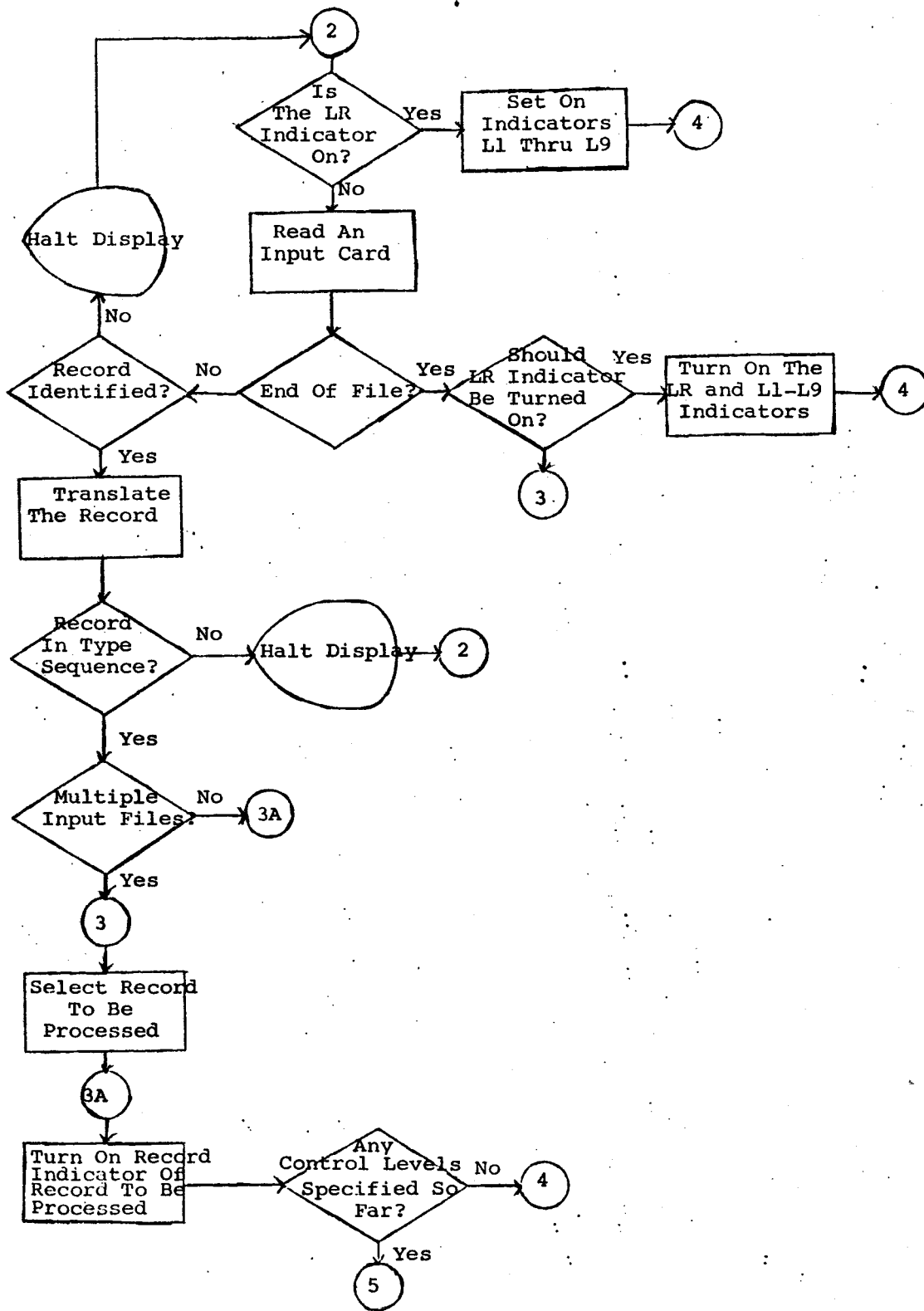


Figure 2-8. Operating Cycle Flowchart (Part 2 of 3)

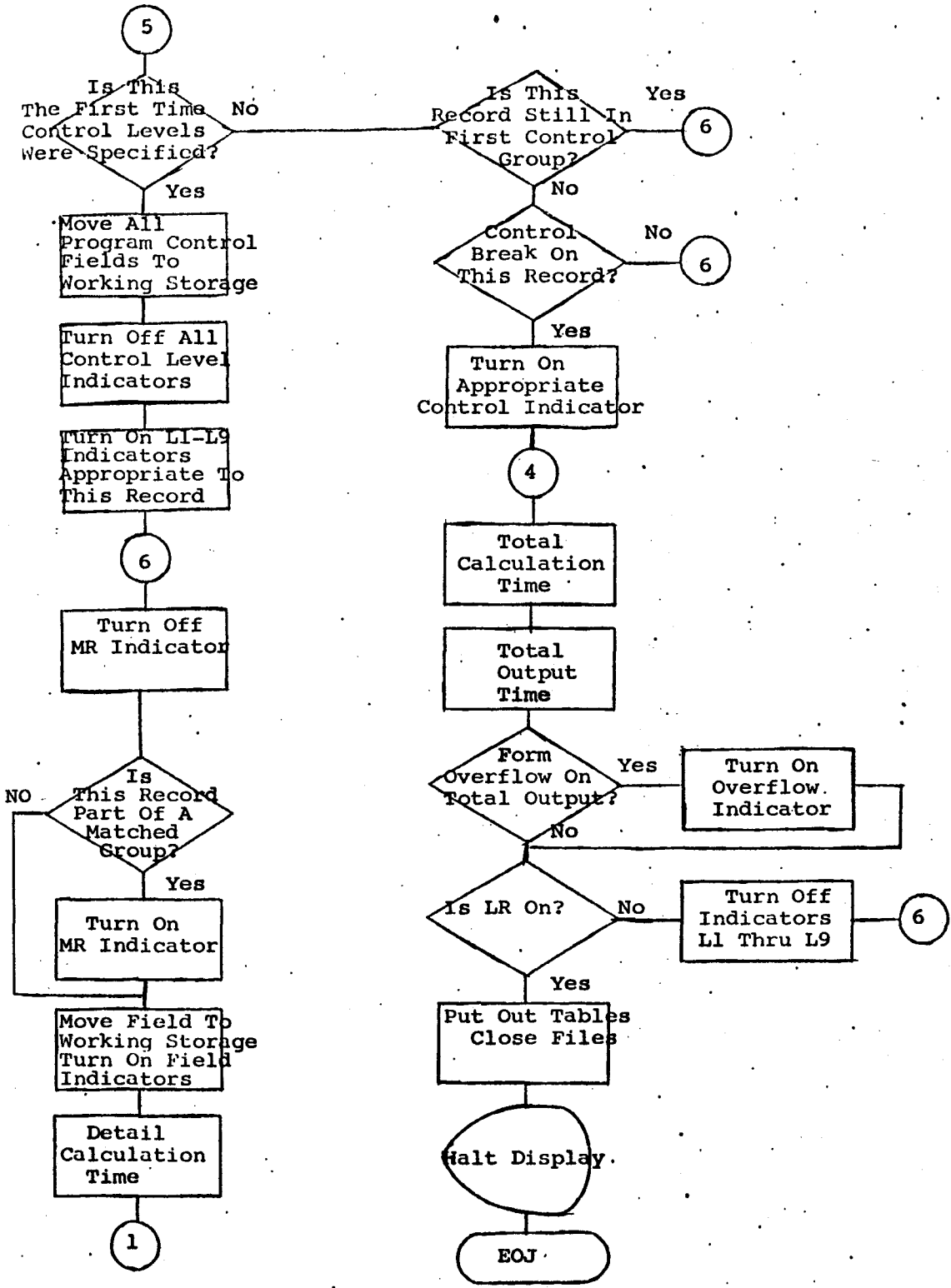


Figure 2-8. Operating Cycle Flowchart (Part 3 of 3)

APPENDIX E. 8411/8414 SORT MERGE

Corrections:

1. When processing the 'SORT' parameter, positional parameter 3, the option of (R) for retaining equal key fields and (D) for deleting equal key fields was not displaying an error halt X'10A2' and the DROC option in the 'OWNCD' parameter was used. The DROC option may only be used with (U) relocation code for handling equal key fields by own-code module.
2. Correction made so the creation date of the OLB statement will be checked with the expiration date of the output tape or disc.
3. Problem resolved to allow halt X'1026' to occur (INVALID BLOCKING FACTOR) if a one digit blocking factor was given. Results previously were unpredictable.
4. The addresses referencing ILBF, ILBC, ILBG, ILBV, OLB, OLBF, OLBC, OLBG, OLBV, and OLBX labels between own-code and sort have been corrected.
5. Corrected problem where zoned decimal (Z) was specified in the field card. The results were that the field was being unpacked before sequence checked, result being a X'3003' stop (sequence error) for negative numbers.
6. Residue was being lost when getting a record after a block of data was sorted internally. This would only occur for variable type records that had residue and ended at the maximum residue buffer address and the sort tried to write the residue before moving it into the residue buffer.
7. An instruction labeled 'ROPE' was in an invalid position in the coding causing unpredictable errors when merging with own-code options.

Enhancements:

1. The DATE statement will no longer be required by the SORT/MERGE and all coding required to process the Date statement has been deleted.
2. General housekeeping of deleting unreferenced coding and labels.

Special Notes:

1. If the number of records inputted to the sort does not equal the number of records outputted by the sort, a halt X'6EEE' will result. If "1" is keyed into location 4, the halt will be ignored. No keyin or any keyin other than 1 will cancel.
2. To alleviate the problems caused by the restriction that only the last reel of tape input, in a multivolume blocked sort can have a short last block, OWNCODE IPRO (Input Processing) modules are being supplied.

I?SL Standard Labels
I?NS Non-standard Labels
I?NL Non-Labeled

Parameter cards are prepared the same as a conventional sort with the inclusion of an "OWNCD" card, (See Example 1 on page 3 of 3). Input block sizes cannot be larger than 1800 decimal bytes. For block sizes 900 bytes or less two I/O areas for faster inputting can be requested by specifying a "W" in positional parameter 11 of the "IN" card. (See Example). Halt X'39uu' will occur to signal that the next input reel should be mounted. Alternate input units can be specified.

3. When specifying the memory size (16383, 24575, 32767) for the 'SORT' statement and DUMP=RES is specified when the supervisor is generated, take into account the size of the resident dump (approximately 300₁₀ bytes) and subtract it from the memory size.

Software Restrictions:

Permanent Restrictions:

1. The first byte of an input record may not contain X'FF'.
2. The 'D' for deleting equal key fields may not be used in the 'SORT' parameter is OPRO (own-code) is used. The DROC own-code for data-reduction can not be used with OPRO. Manual change will be made to reflect above restrictions.

//

```
EXEC PHASEZRO
DATA T
OWNCD { I?SL FOR STANDARD LABELS
        I?NS FOR NON-STANDARD LABELS } .....
        I?NL FOR NO LABELS
DISCS 00, SORTWRK1, 01, SORTWRK1.
ILB    SORTOUT1, 69321, 0001, 01.

IN     S, SORTOUT1, 05, 06, T, 03, 10, 0080, .. { S
                                                W IF INPUT BLOCKSIZE 900 BYTES OR LESS } ..
                                                (TWO I/O AREAS IF 'W' USED)

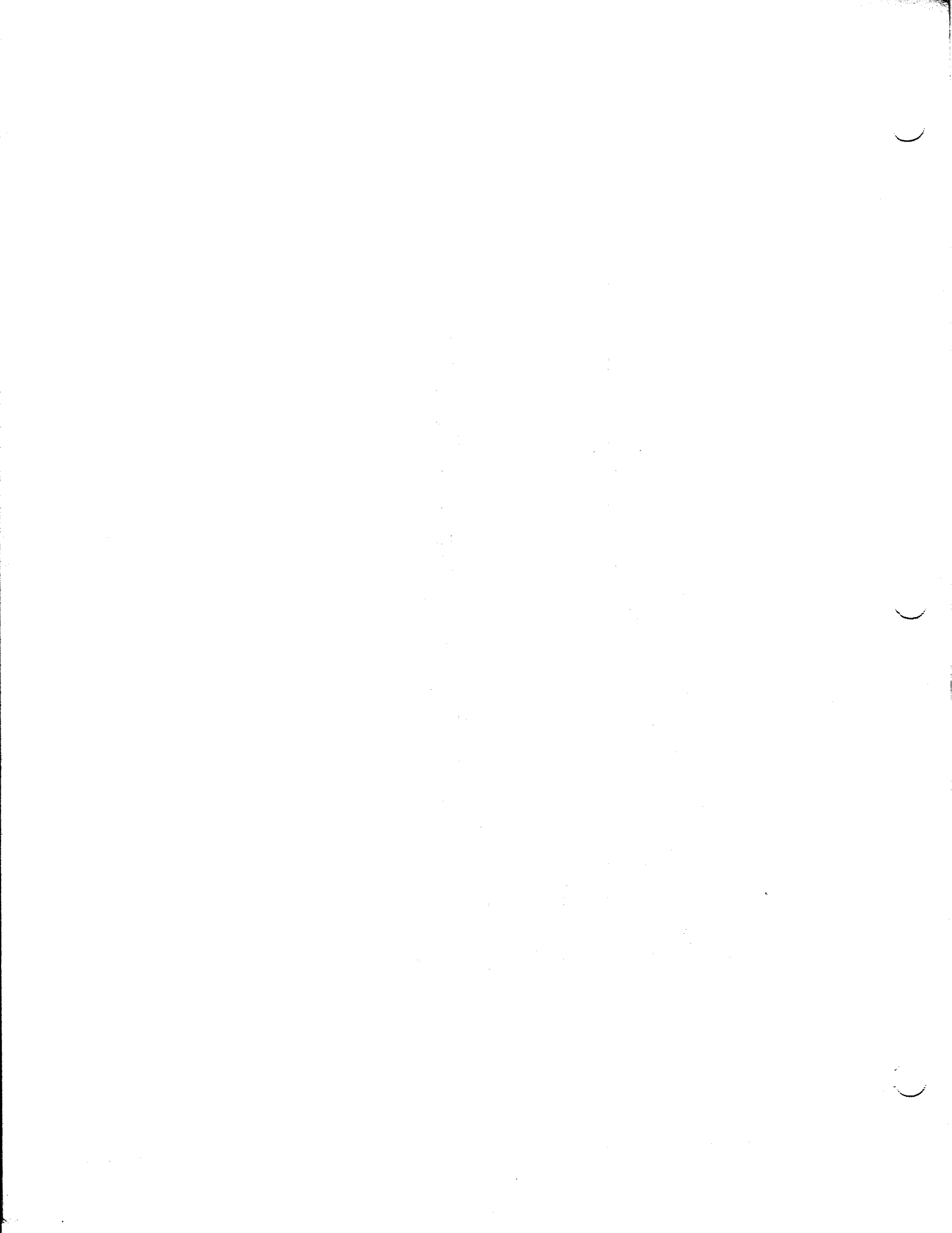
OUT    07, 07, T, 10, .., R.
SORT   30000, R, R, .., D, N.
FIELD  CA0005005.
END
```

/*

```
DATA C
OWND 00 SYSFILE
      { I?SL FOR STANDARD LABELS
        I?NS FOR NON-STANDARD LABELS } ] REPLACE WITH RELOCATABLE DECK
      I?NL FOR NO LABELS ] IF OWNCODE NOT ON DISC
END
```

/*

EXAMPLE



APPENDIX F. Implementation of DNCOS in 16-K memory.

Under certain circumstances, it is possible to implement 8411/8414 DNCOS on a 16-K machine. The discussion below is intended to be used for guidance of customers and UNIVAC personnel who may be considering such systems in the future. Any decision to attempt installing a 16-K DNCOS system should be made only after a thorough review of the needs of the particular installation and after assembly or compilation of "worst case" programs.

1. SYSGEN Limitations. It is not possible to perform a system generation at a 16-K installation using the master discs supplied by UNIVAC. It is recommended that the generation be performed at another installation. If this is impossible, the following guidelines may be helpful:
 - a. The elements on master disc #1 may be copied to another pack, replacing all loadable elements from "SUPR" through "LIBS4" with similar loadable elements from the 16-K site's system pack. Phase I and the linking of utilities may then be performed.
 - b. The loadable R11T compiler from disc #1 may be copied to the pack containing other elements generated during Phase I. This compiler is linked at X'1380'. It is impossible to link a compiler from the relocatable elements supplied since the Linker does not have sufficient memory space in a 16-K machine to link the largest phase of the compiler.
2. Generated Supervisor Size. While DNCOS has been operated successfully with previous Supervisors as large as X'1200', it is naturally desirable to generate the smallest possible Supervisor. The following should be noted to assist in accomplishing this:

- If the standard (0711) reader is used for the control stream, and if not more than three disc drives are specified by the UNT1 parameter, and,
- If the highest logical unit number in the configuration is 10 ("CFG0A"), the size of the generated Supervisor will be less than X'1020'.

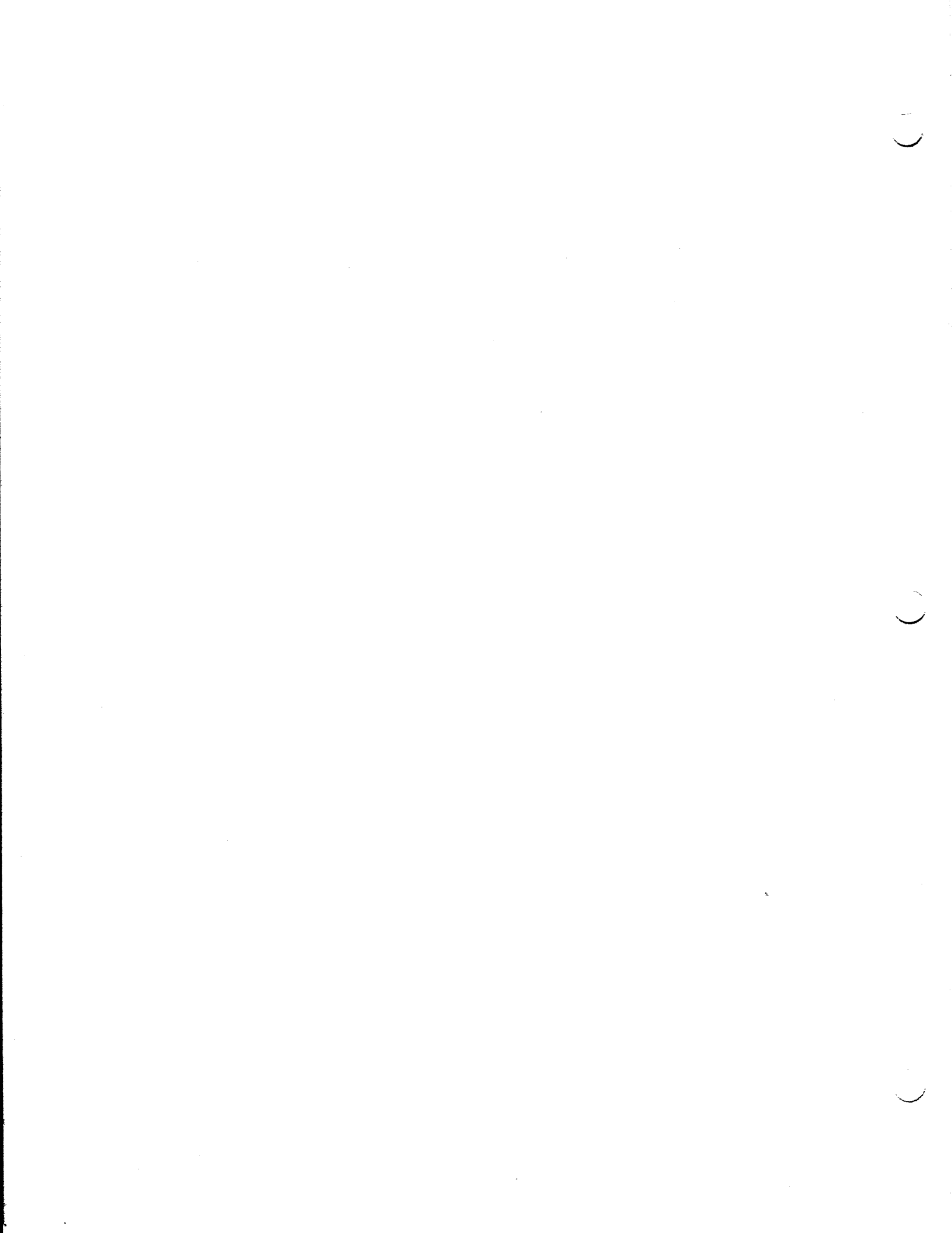
This will enable Dump/Restore to fit into memory with 64 (X'40') bytes to spare. DPRS is the largest utility supplied with this DNCOS release. If the user wishes to save additional memory by using non-standard logical unit numbers (i.e., assign L.U. #3,4 and 5 to reader/printer/punch), it will be possible to keep Supervisor size below X'1000'. It is not a recommended practice to link programs immediately above the Supervisor as it cannot be predicted whether future changes to the Supervisor may cause it to increase in size. While Linker, Dump/Restore, and LIBS should be linked very low, it would probably be feasible to link an RPG compiler higher (or use the prelinked compiler to allow for possible future Supervisor expansion.

3. Programming Techniques. Needless to say, it is not possible to write as large a program for a 16-K machine as for a larger system. However, a certain amount of finesse will enable the user to squeeze surprisingly large programs into 16-K. Some general guidelines follow:

- a. Keep block sizes within reason. For RPG programs which handle several files at a time, 500 bytes has been found to be a good rule of thumb. Where possible in Assembler programs, use the same work area to process records from various files.
 - b. In RPG programs, make the maximum use possible of the asterisk enhancement where more than one line of code is governed by the same indicator(s). Use GOTO and TAG statements to branch around calculations that would involve many tests of indicators that could be avoided.
 - c. In Assembler, make any initialization routines which are only used once overlayable by putting them in work areas (DS).
 - d. In Assembler, where possible, use DTFCZ and DTFPZ in place of DTFC and DTFP (See Appendix G).
 - e. Make GET/PUT/READ/WRITE transient for SAM and ISAM files wherever possible. While throughput may seem to suffer by repeated accesses to TRANSCYL, if transientizing all disc file processing functions is the only way to make the program fit into available memory, the throughput achieved will probably be much better than that obtained by breaking the program into separate programs and making multiple passes through the same files.
 - f. SAM and ISAM processing have been mixed successfully in the same RPG programs but you will run out of memory much sooner with both types of processing in one program than with only one of the two.
4. LIBS. In a 16-K machine, the only version of the Librarian which will fit into memory is one which is linked from the relocatable elements LIBSØ1 or LIBSØ2. If LIBSØ2 is used, it must be linked no higher than X'116Ø' if it uses the serial punch; X'1Ø4Ø' if it uses the row punch.
 5. Tapes. It is extremely unlikely that a DNCOS system which included magnetic tapes would be functional in 16-K, even using the non-resident Tape Dispatcher.
 6. COBOL. The COBOL Compiler will barely fit into 24-K. There are no plans to make it operable in 16-K. However, there is nothing to prevent the compilation of COBOL programs on larger systems for use on 16-K machines. As in Assembler and RPG, much more finesse must be employed in a smaller memory size.
 7. The FORTRAN compiler can be used in 16-K but DATA and EQUIVALENCE statements are not accepted by the compiler. The use of large arrays should be avoided, where possible, to enable the generated program to fit. Since the largest part of any loadable FORTRAN program is usually the automatically-included library subroutines, it may be desirable to analyze the external references generated by the FORTRAN compiler to determine whether the number of library subroutines required by the program can be reduced. It may also be necessary to employ segmentation (through CALL LOAD and CALL FETCH statements) to make a very large program fit into minimum memory.
 8. Linker. For the Linker to be operated in 16-K memory, it must be linked no higher than X'1ØBØ'.

9. Miscellaneous Sysgen Considerations:

When linking the RPG compiler with the Supervisor and Linker which will be used on a 16-K system, it will be necessary to alter location E?TM (see Supervisor assembly listing) to reflect the memory size of the larger machine on which the linking is being done. (X'3F' should be changed to X'5F' or X'7F' after IPL but before executing the Linker. A PARAM R card could be inserted ahead of the EXEC D111 card for this purpose.)



APPENDIX G. TRAN AND TRANSIENTS

1. The following changes or improvements have been made to TRAN:
 - a. When TRAN finishes writing out a track of system transients, it prints the number of bytes remaining on the track. Each time it writes out a user transient, it prints the number of bytes remaining on the track. This information may be useful to installations which maintain their own transients.
 - b. TRAN always verifies that the transient it is about to write out will fit in the space remaining on the track. If it will not fit, X'74BB' is displayed and the job is cancelled.
 - c. TRAN produces the following new halt displays:

74EE - TRANSCYL cannot be located on the output unit. Press START to retry the search.

74FF - The relocatable module which was to be written out as a transient cannot be located on the input unit. Press START to retry the search.

Each halt display is preceded by a self-explanatory printout. The locator/loader stops which TRAN formerly caused to occur are no longer given. The printed message includes the locator/loader stop that the Supervisor would have produced.
 - d. When TRAN is asked to write out a user transient, if the "base address" is omitted from the input card (columns 34-37), TRAN will assign the address of the system transient area as the transient's base address.
 - e. TRAN now prints, at the top of its printer output, the Volume Serial Number, device address, and logical unit number of the system-resident disc pack.
2. There have been several additions to the UNIVAC-supplied system transients as well as changes in module names and track assignments. Therefore we are including herewith a complete list of transients along with such additional information as might be necessary to an installation which needs to replace standard transients with their own.

<u>I.D.</u>	<u>Module Name</u>	<u>Description</u>
0001	none	Supervisor locator/loader routines. TRAN writes out the contents of the system transient area. At IPL time, this area contains these routines which were assembled as a part of the Supervisor.
0002	UQ?OPCL	Inquiry IOCS OPEN and CLOSE routines. This transient is called from the resident portion of the inquiry IOCS generated by the DTFUQ macro. Necessary only in OS-500 system with inquiry capabilities.
0003	UQ?GTPT	Same as 0002, but contains inquiry IOCS GET, PUT and CNTRL routines.

<u>I.D.</u>	<u>Module Name</u>	<u>Description</u>
Ø1Ø1	SAM?OPN1	Sequential access method (SAM) OPEN Part One. Called by SAM IOCS to OPEN sequential disc files.
Ø1Ø2	S?QCLOSE	SAM CLOSE. Called by SAM IOCS to CLOSE sequential disc files.
Ø1Ø3	IS?USETL	Indexed sequential access method (ISAM) unblocked SETL. Called by ISAM IOCS in response to issuance of SETL macro.
Ø2Ø1	IS?MOPEN	ISAM OPEN. Called by ISAM IOCS to OPEN indexed sequential files.
Ø2Ø2	IS?CLOSE	ISAM CLOSE. Called by ISAM IOCS to CLOSE ISAM files.
Ø2Ø3	RPGA??DD	ISAM ADD CCW generator. Called by RPG compiler to generate CCW chains for programs which will add to ISAM files.
Ø3Ø1	IS?GETPT	ISAM GET and PUT. Called by ISAM IOCS for sequential processing of ISAM files.
Ø3Ø2	IS?HIADD	ISAM add high. Called by ISAM IOCS to add records whose keys are higher than those existing in the file.
Ø3Ø3	IS?RDWRT	ISAM READ and WRITE. Called by ISAM IOCS for random processing of ISAM files.
Ø4Ø1	S?QTTGET	SAM GET. Called by SAM IOCS in response to GET macro.
Ø4Ø2	S?QTTPUT	SAM PUT. Called by SAM IOCS in response to PUT macro. Cannot process undefined or variable-length records.
Ø4Ø3	COM?TRAN	Communications routines. Called by Supervisor in response to SRC's 30, 32, 36 and 38 and also by "Comm Cleanup" (E?CU).
Ø5Ø1	T?PEOPEN	Tape OPEN. Called by Magnetic Tape IOCS to OPEN tape files.
Ø5Ø2	T?PCLOSE	Tape CLOSE. Called by Tape IOCS to CLOSE tape files.
Ø5Ø3	T?PTTGET	Tape GET. Called by Tape IOCS in response to GET macro.
Ø6Ø1	IS?BADIN	ISAM blocked add/insert. Called by ISAM IOCS to insert new records in ISAM files whose record format is blocked.
Ø6Ø2	IS?UADIN	ISAM unblocked add/insert. Same, but for unblocked records.
Ø6Ø3	IS?BSETL	ISAM blocked SETL. Same as Ø1Ø3, but for blocked records.
Ø7Ø1	T?PTTPUT	Tape PUT. Called by Tape IOCS in response to PUT macro.
Ø7Ø2	IS?INDEX	ISAM index-in-memory. Called by ISAM IOCS as a part of OPEN processing to bring an ISAM file's cylinder indices into memory.
Ø7Ø3	DISC?IPL	Disc IPL block. Called by LIBS in response to a ▽ BOOT command. This transient contains a copy of the standard 8411/8414 IPL block.
Ø8Ø1	RPG?WEC2	Generation transient. Called by RPG compiler to open the library file to which the generated program will be written.
Ø8Ø2	RPGWED3	Generation transient. Called by RPG compiler to write out the generated program.
Ø8Ø3	RPG?WECL	Generation transient. Called by RPG compiler to close the object file.

<u>I.D.</u>	<u>Module Name</u>	<u>Description</u>
Ø9Ø1	RD??PRNT	Printer/control stream transient. Called by IOCS routines generated by DTFPZ and DTFCZ.
Ø9Ø2	T?PETRAN	Transient tape dispatcher. The generation of this transient is an option at system generation time. It is only used if Supervisor and Job Control are generated with TTRN=YES specified. It is not loaded into the system transient area but into the area between the Supervisor and Job Control. Its size varies according to the options selected when generated. It is loaded into memory by Job Control when a program is to be executed which will use magnetic tape units, as defined in its program-i.d. record.
ØAØ1	FILE?	File and Source Element Locator. Called by LIBS, COBOL, RPG, Job Control SJCS routine, and Cross Referencer to locate a library file and/or source element.
ØAØ2	SAM?OPN2	SAM OPEN Part Two. Called by SAM?OPN1 to complete OPEN processing.
ØBØ1	Z\$DISC	FORTTRAN disc OPEN. Called by FORTRAN Library routine Z?DS to open disc files used by FORTRAN-generated programs.
ØCØ1	D?H?OPEN	Direct access method (DAM) OPEN. Called by DAM IOCS to OPEN direct access files.
ØCØ2	D?H?CLOS	DAM CLOSE. Called by DAM IOCS to CLOSE DAM files.
ØCØ3	D?H?KEY	DAM Key processing. Called by DAM IOCS to perform Key processing as a transient.
ØCØ4	D?H?ID	DAM ID processing. Called by DAM IOCS to perform ID processing as a transient.
ØCØ5	D?H?AFTR	DAM AFTR processing. Called by DAM IOCS to perform AFTR processing as a transient.
ØCØ6	D?H?IORG	DAM IORG processing. Called by DAM IOCS to perform IORG processing as a transient.
ØDØ1	FAST?LDR	Fast locator/loader routines. Called by Supervisor to locate and load programs in the system execution area (SYSEXAR).
ØDØ2	RIRO?TRN	Roll-in/roll-out transient. Called by OS-500 Supervisor in response to an inquiry request when it is necessary to roll out the main program prior to searching for an inquiry program. Called again by the Supervisor to roll in the main program after the inquiry request has been processed.
ØDØ3	PARAM?	PARAM card processing. Called by loader to verify PARAM cards and apply patches contained therein to the program about to be executed. In OS-500 system, if Job Control was generated with CLOG=ALL specified, PARAM? prints PARAM cards on the system console. If the current control stream is on disc, PARAM? causes a return to the card reader if it detects a "/ DATA READER" statement.

3. The following information is provided concerning the bar printer and control stream reader transient IOCS macros, DTFPZ and DTFCZ respectively:
 - a. DTFCZ is a macro which generates the resident shell of a control stream reader routine. Users who program in assembler language may call DTFCZ in place of DTFCS in order to save approximately 270 bytes. DTFCZ has one key-word parameter, EOFA, whose meaning is identical with the EOFA parameter in DTFCS. Consult the current edition of Operating System IOCS (UP-7526) for information concerning DTFCZ.

- b. DTFPZ is a macro which generates the resident shell of an IOCS routine for the bar printer. Users who program in assembler language may call DTFPZ in place of DTFPR in order to save approximately 500 bytes. DTFPZ has the following keyword parameters:

BKSZ, CNTL, DEVA, FONT, OTBL, PRAD, PROV

The meanings of all parameters are identical with those of the same parameters in the DTFPR macro. UP-7526 also contains complete information on DTFPR. DTFPZ has no buffered print capabilities.

- c. Either or both macros may be called within an assembler language program or may be used to produce relocatable elements which can be linked to a main program. The user must provide START and USING statements in either case, just as he would do with DTFCS and DTFPR.
- d. The routines which are generated by the DTF CZ and DTFPZ macros call in a standard system transient, RD??PRNT which occupies Head 9, Record 1.
- e. Although the use of these IOCS routines, like that of any other routines that use transients, will result in a savings of main program size, the same considerations of program size vs. throughput apply. A program which does a large amount of printing and card reading and which also uses Tape and Disc IOCS transients heavily would be slowed down noticeably by having to load a new transient each time a card was to be read or a line printed. A program which used other transients only for opening and closing tape and disc files would not be slowed at all by transientizing reader/printer functions.
- f. A program which requires both the control stream reader and the bar printer need not transientize both functions. The "standard" macro could be used to generate the IOCS routine for the high volume file and the new macro for the low volume file. (DTFCS plus DTFPZ, or DTF CZ plus DTFPR).
- g. Modules produced by these new macros are supplied for linking to all UNIVAC-supplied utilities and language processors, except for the RPG and COBOL compilers, whose reader/printer routines are internal, and TRAN, which must be able to operate in the absence of any transients.
4. The following information is included in this release memo in response to many questions received in Blue Bell concerning TRAN and transients. It is hoped that it will clear up any misconceptions that might exist on the subject.

Q - Why is it necessary to run TRAN with one Supervisor as opposed to another?

A - When TRAN is asked to write out system transients, it ascertains the address of the system transient area. It then locates each relocatable element whose program-i.d. is on a list which is within TRAN itself. The relocatable addresses within each element are converted to absolute addresses which are dependent on the location of the system transient area. The element is then written to TRANSCYL as a single block. If the contents of TRANSCYL are then copied to a pack which contains a different Supervisor, or if the Supervisor under whose control TRANSCYL was written is replaced, the transients will not function correctly if the Supervisor with which they are now used has a system transient area at a different location.

- Q - How can we ensure that we have the "right" transients on a given pack?
- A - One of the recent improvements to TRAN was the addition of a header line which contains the VSN, LU, and device address of the pack from which the Supervisor was booted. This should be checked to insure you booted from the right disc unit. Also, Job Control checks TRANSCYL each time it (JBCN) is loaded to see if it can read the Locator/Loader Transient and to compare an address constant within that transient against the address of the system transient area. If the transient is unreadable or the address does not match, a X'41CØ' display is given. This display may be expected when a newly-created system pack is booted; if it occurs under any other circumstances, it indicates that TRANSCYL is unusable and should be rewritten.
- Q - Why can't we copy TRANSCYL with LIBS?
- A - TRANSCYL is not a library file. LIBS can process only those files which are in library format, i.e. two extents and 160-byte blocksize.
- Q - If TRANSCYL is not a library file, how can we copy our transients to a new system pack?
- A - We recommend the use of the Dump/Restore Utility (DPRS). Use the file-to-file dump feature. TRANSCYL may even be located on different cylinder(s) on the new pack. It only takes a few seconds for DPRS to copy a file as small as TRANSCYL from pack to pack. If tape units are available, TRANSCYL may be dumped to tape along with other files for backup purposes. A procedure which would be more time-consuming and disc space-consuming would be to preserve all the relocatable transient elements from the release disc/tape and rerun TRAN each time you create a new system pack.
- Q - If we use the Transient Tape Dispatcher feature, can we still use the Tape IOCS transients? Do we have to use them?
- A - There is no relationship between the use of the Transient Tape Dispatcher and the use or non-use of Tape IOCS transients. Tape IOCS transients are loaded into the system transient area; the Transient Tape Dispatcher is loaded into the area between the Supervisor and Job Control. The Transient Tape Dispatcher remains in memory until Job Control is recalled by the CANCL or EOJ macro. It is reloaded by Job Control each time a program is to be executed which needs tape drives.
- Q - If we are using the Transient Tape Dispatcher, do we have to link our compilers high enough so it won't conflict with them?
- A - If you link anything that doesn't use tapes during its own execution at an address above the Transient Tape Dispatcher, you are defeating the purpose of generating and using this feature. R11T, KSMB, DL11, and FORTRAN do not use tapes; COBOL can read its source from tape but if you linked the COBOL compiler at the higher of the two possible link addresses, you would then have to ask it to produce relocatable code for programs that did not use tapes so that you could link them at the lower address. The Transient Tape Dispatcher feature was designed for those installations whose use of tapes is intermittent or limited to a small number of programs or application areas. If you use the feature, you must take extra steps (providing UNITS cards, specifying higher base addresses, generating relocatable code) to create a small number of programs. If you find that you are taking those extra steps for too many programs, you may wish to regenerate your system without this feature.

Q - If we want to put our own transients on TRANSCYL for loading via SRC 44, how do we get them onto TRANSCYL? And where?

A - See paragraph 2 above for a list and description of all UNIVAC-supplied transients. You may see several transients whose functions are unnecessary in your system, e.g. a configuration that doesn't include magnetic tape units doesn't need the "T?" transients. You may wish to "deceive" TRAN by giving your own transients the names of unnecessary transients, then replacing the UNIVAC-supplied transients with your own in the library file which is input to TRAN. Another approach would be to let TRAN write out all UNIVAC-supplied transients, then overlay our transients with yours by telling TRAN to write out specific transients. It must be remembered that the writing of a new transient causes the erasure of any other transients on the same track which have the same, or higher, i.d. For instance, if you overlaid R1RO?TRN (i.d. 0D02) with your own transient, you would then have to rewrite PARAM? (i.d. 0D03) since the latter transient is used for processing PARAM cards. 8414 users may wish to take advantage of their greater track capacity to write their own transients after UNIVAC-supplied transients on each track.

Q - What is the difference between a "standard" or "system" transient and a "user" transient?

A - To the Supervisor, a "user" transient is any transient which is called for by an SRC 46. This kind of transient may be loaded from any cylinder, including TRANSCYL, and into any memory location, including the system transient area. This feature is only available if UTRN=YES was specified when the Supervisor was generated. Any transient which is called by the SRC 44 is considered a "standard" transient and must be loaded from TRANSCYL into the system transient area (tag UT?? on Supervisor assembly listing).

To TRAN, a "standard" transient is any transient which is to be written out in response to an input card which has a blank in column 16. When TRAN reads such a card, it writes out, either to TRANSCYL or to a user-specified cylinder, all elements which appear on a predetermined list. To TRAN, a "user" transient is any element whose name appears in columns 16 to 23 of an input card. This element may or may not also appear in TRAN's list of "standard" transients; it may or may not be written to TRANSCYL. TRAN makes no attempt to verify whether the user-specified output cylinder is assigned to a file or to examine its contents.

Q - Do we have to reassemble or recompile all our programs every time a new DNCOS release comes out?

A - Reassembly or recompilation should only be done if necessitated by some correction or improvement announced in the release memo. In general, old programs (i.e. which contain previous levels of IOCS routines) can be run with new transients since every effort is made to avoid making changes to transients which force reassembly or recompilation of user programs. However, you should not attempt to use new programs (i.e. with resident IOCS routines from the new release) with transients issued with previous releases.

APPENDIX H. SUPERVISOR AND JOB CONTROL

SUPERVISOR

For changes which affect only OS-500 users, See Appendix M (OS-500).

The following enhancements have been made:

1. The restriction on the use of the System Job Control Stream feature where the Control Stream Reader is the 1004 reader has been lifted.
2. It is now possible, while processing a control stream from disc, to return to the "read" reader for control stream data input. See Appendix B (SJCS) for details.
3. The size of the system transient area has been reduced to X'520' (1212₁₀) bytes for all systems except for certain OS-500 systems (see Appendix M.) This reduction in size will result in a smaller Supervisor for most configurations.
4. All processing of PARAM cards is now done in the transient PARAM?. This transient verifies the content of each PARAM card and produces the following halts in error situations:

41CA - In a "normal" PARAM card, commas did not appear between relative location and length or between length and inserted data. In a PARAM R card, commas did not appear between each two bytes of patch material or the patches did not terminate with a blank column.

41CB - The address at which data would be inserted by either kind of PARAM card is within restricted memory (less than X'0040'), higher than highest address as defined in boundary table, or in a PARAM R card is odd.

41CC - A character other than a valid hexadecimal digit was read where only hexadecimal digits 0 through F were expected.

After halt, the reader may be cleared and a corrected card placed in the hopper to resume processing. If the control stream is being read from disc, pressing START will cause the next image in the system job control stream file to be made available.

5. The ability to use PARAM R cards is now a standard feature in all 8411/8414-resident systems. Since all PARAM processing is done by a transient, this feature is available at no extra cost in Supervisor memory size. The REP parameter no longer needs to be specified to obtain PARAM R capability and has no affect on Supervisor generation.
6. The disc dispatcher formerly produced 69u4 stops for command reject or invalid command errors as well as for a track overrun. The two situations are now distinguished from one another by providing a 69u5 stop for an invalid command or command reject.
7. The disc dispatcher sometimes returned an incorrect address in bytes 2-3 of the disc request packet or failed to switch tracks properly when a multitrack read or search chain encountered a correctly-flagged defective track at the end of a cylinder. This discrepancy has been corrected.
8. The Supervisor locator routines are used by Job Control and TRAN to locate files and/or elements. This sometimes resulted in misleading 40xx or 14xx halts. Options have been incorporated into the locator to return control to the calling program when such errors occur. See TRAN and JBCN information elsewhere in this release memo for information concerning new halts.

JOB CONTROL

For changes which affect only OS-500 users, see Appendix M.

The following enhancements have been made:

1. At IPL time, if the system resident disc pack does not contain a file named TRANSCYL, Job Control will produce a 41C1 display. If START is pressed, the locator error message is given which describes the reason for the failure to locate TRANSCYL (i.e. 14xx or 40xx). This error is unrecoverable, except by running DASM under another Supervisor to allocate TRANSCYL, then rebooting from the pack on which the error occurred.
2. Each time Job Control is loaded, it reads the beginning of the locator/loader transient to compare an address constant against the address of the system transient area which is available at location X'0156'. If a read error occurs or if the address is not correct, a 41C0 display is given. If START is pressed, Job Control will begin processing the control stream. The halt is to be expected when booting a newly-created system. If it occurs at any other time, it indicates that TRANSCYL is unusable and that TRAN should be re-executed.
3. The CSR keyword has been deleted from those keywords which may be specified when generating Job Control.
4. Job Control now resets the interrupt and SRC table entries of all devices (other than tape, disc and 1004) which were allocated to a main program. This prevents possible errors caused by attention interrupts from devices such as paper tape units and 0716 readers.
5. When Job Control is in a FINIS loop and job cancellation takes place through X'F0' keyin or through CANCL macro issued within the Supervisor or a symbiont, Job Control is reloaded but does not attempt to restart the control stream. Control stream restart is accomplished only through the X'E0' keyin.
6. An error in Job Control's cancel routine has been corrected which formerly caused a PROC ABN if a DVCDN or DVCUP card was encountered after cancellation.
7. The routine which scans the operands of a Job Control statement has been corrected to accept a maximum of eight characters per operand.
8. See other Appendices of this release memo for descriptions of the NOTE and VOL statements, SKIP statement enhancements, and the implementation of the System Execution Area.

WARNING:

The Supervisor and Job Control Procs distributed with this release are for use in generating an 8411/8414-resident system and have not been tested as tape-resident or 8410-resident systems. Any use of the Procs to generate such systems is at the programmer's own risk.

APPENDIX I. 8411/8414 Disc Assembler (KSMB)

A. Enhancements

ISEQ directive can now be placed anywhere in the source statements and will sequence check till End record.

1. New Instructions.

a. Macro Definition Exit (MEXIT)

The macro definition exit (MEXIT) statement may be used only within a macro definition that is written in Macro format. This statement indicates to the assembler that the processing of a macro definition should be terminated before ending normally with an END statement.

The format of the MEXIT statement is:

<u>Label</u>	<u>Operation</u>	<u>Operand</u>
unused	MEXIT	unused

When MEXIT is used, the assembler terminates processing the macro definition and processes the statement in the source program following the macro call instruction that called the macro definitions containing MEXIT.

b. PNOTE Message Statement (PNOTE)

The PNOTE statement may be used to generate comments or error messages on the printer listing and may be used in a macro or at the source code level.

The format of the PNOTE statement is:

<u>Label</u>	<u>Operation</u>	<u>Operand</u>
unused	PNOTE	* or 'E', 'C'

where:

* indicates an asterisk to be printed in the error flag field

'E' is an error character string enclosed in apostrophes

'C' is a comments character string enclosed in apostrophes

The PNOTE statement indicates to the assembler that a comment line is to be generated in the output listing.

c. Conditional branch (IF)

The conditional branch (IF) statement conditionally alters the sequence of source statement processing.

The format is:

<u>Label</u>	<u>Operation</u>	<u>Operand</u>
symbol	IF	e, S _L

where:

symbol is a sequence symbol

e is a logical expression

S_L any symbol which is a LABEL directive

If, after the logical expression has been evaluated, the condition is true (a value of 1), you branch to the LABEL directive that is specified by the symbol in the operand. If the condition is false (a value of 0), the statement in the source code following the IF statement is permitted.

Example:

	10	16
<u>Label</u>	<u>Operation</u>	<u>Operand</u>
1 TEST	IF	(&SIZE>10), JOHN
2	MNOTE	'INVALID LENGTH'
3	MEXIT	
4 JOHN	LABEL	

If the value of &SIZE(1) is greater than ten, the next statement to be processed is JOHN.

Caution must be taken when specifying the LABEL if outside a DO and ENDO nest.

d. MNOTE Message Statement is:

<u>Label</u>	<u>Operation</u>	<u>Operand</u>
unused	MNOTE	'm' , 'm' s, 'm'

where:

'm' is a message enclosed in quotes.

s is a severity code. (The severity code is for compatibility only and is ignored by the assembler).

When MNOTE is used, the assembler processes the message in the operand field of the source code statement excluding the enclosing apostrophies.

Example:

	10	16
<u>Label</u>	<u>Operation</u>	<u>Operand</u>
1	DO	CHNL=0
2	MNOTE	'CHANNEL IS MISSING!!!'
3	MEXIT	
4	ENDO	

If the DO directive statement is true, MNOTE will be generated and the MEXIT operation then causes the assembler to terminate the macro definition.

e. Macro Instruction (OPR)

Problem programs can issue messages that are printed on the system console with the option of allowing replies from the operator. Assembler will now handle this macro instruction. See OS-500, Appendix M.

2. A special form of the comments statement is also available. This form is used to include comments that are not to be generated in the output listing. This type is written with a period in Col. 1 of the assembler coding form, followed by an asterisk (*) in Col. 2, followed by the comment.

Example:

. * This is an internal comment, not printed.

3. If the remainder of characters to be generated from a 'DC' constant, which contains more than eight characters and the eighth character is identical to the remainder of data, they will not be printed. To have the remaining characters printed, the PRINT directive must be used. The option must be ON; ON,GEN; or ON,NOGEN.

B. Corrections:

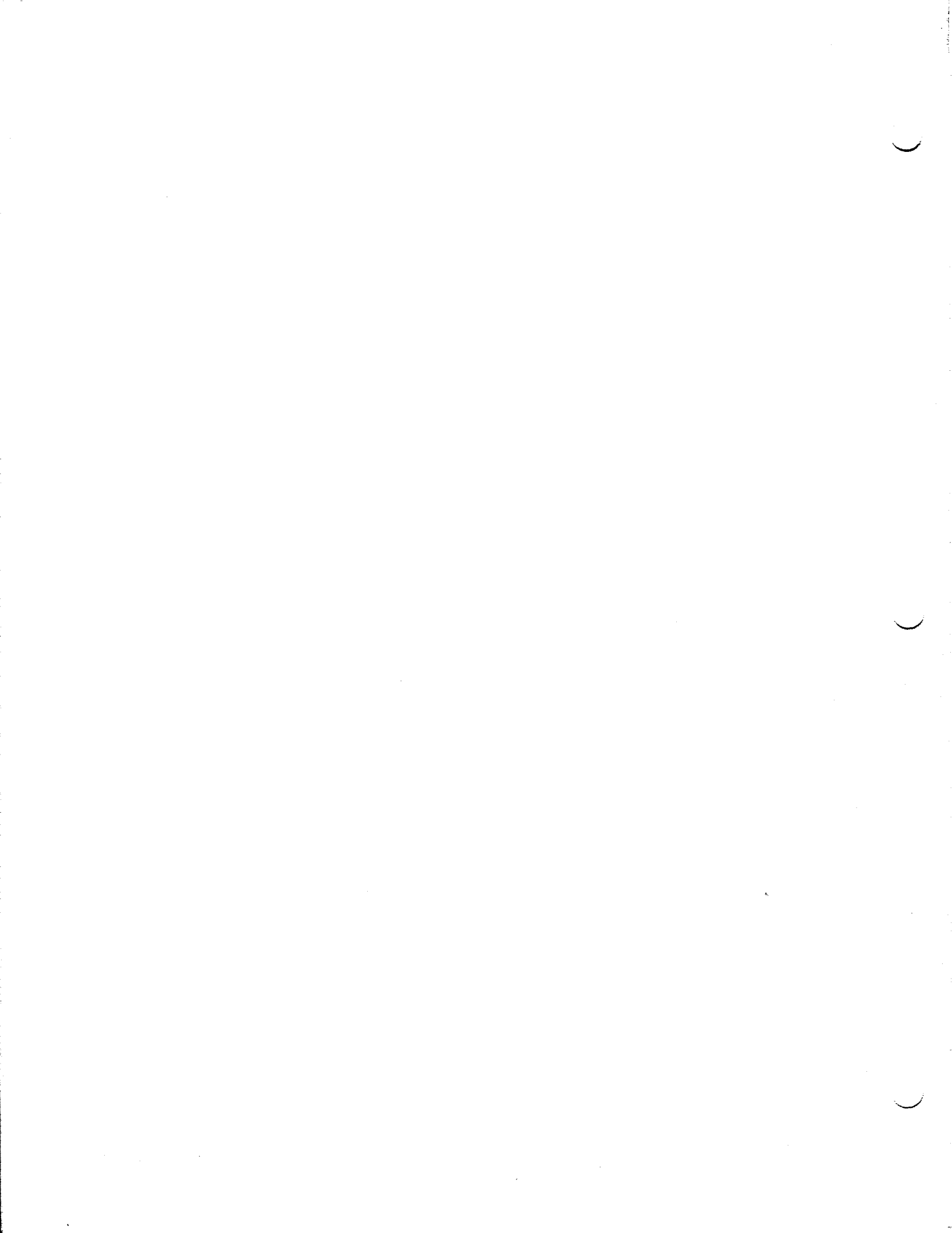
1. Change made to allow the disc assembler to end on a system type record (5500) if the END statement is missing when processing source from disc.
2. The disc assembler will now flag an error if the second operand of the READ, WRITE, and SETL macro instruction is missing.
3. Symbol table was being overlaid when using the restart print option causing a PROC ABN. Problem was corrected.
4. Previous changes for resolving problems when processing 'DC' constants caused the length attribute (L') to be incorrect and flag as an error. This situation has been corrected.
5. If the 1st character of a label is numeric it will not be flagged as an error as stated in previous release. Documentation will be corrected to reflect this situation.
6. Previous problem with ISEQ has been corrected. Line number was not being assigned to last Proc Parameter.

C. Special Restriction

A problem exists when literals are used in programs consisting of DSECTS. To avoid unpredictable results, it is suggested that the LTORG statement be used whenever literals are present, until this problem is resolved.

NOTE:

The number of CSECTS and DSECTS that can be assembled is limited to an area consisting of 100₁₀ bytes. Formatted: (L, ESID, LABEL) 1 byte for length of label (L), 1 byte for ESID, and the label. Maximum value being 10 bytes (with an 8 byte label). Maximum number of combinations of CSECTS and DSECTS with 1 character labels would be 33. This area is variable depending on the label length and the number of CSECTS and DSECTS.



APPENDIX J. 8411/8414 Disc Linker (DL11)

Enhancement

Ability to have optional information in the comments field of the output ELT record.

- a. If a comment is specified in the PRGM linker statement up to 21 characters will be placed into the comments field of the ELT Record. The remaining characters will be truncated.
- b. If no comment is specified in the PRGM linker statement, the Supervisor date (Loc. 280₁₀-282₁₀) will be placed in the comments field of the ELT record.
- c. If a comment is specified in the PRGM linker statement and the supervisor date is requested to show in the comments field of the ELT record, the following Parameter must be used:

```
/      PARAM      0001,01,2
```

This will place the date left justified in the comments field of the ELT record and allow a space, then up to 14 characters of the PRGM statement comment will be placed in the remainder of the comment field in the ELT record. The remaining characters will be truncated.

If the suppress print option is required also, the parameter is:

```
/      PARAM      0001,01,3
```

EXAMPLES:

- 1) PRGM TEST,* THIS IS A COMMENT TEST

The above linker statement will result in the ELT record's comment field to look as follows:

```
LOC. BYTE 60=THIS IS A COMMENT TES
```

- 2) PRGM TEST,*

The above linker statement will result in the ELT record's comment field to look as follows if the / DATE '031075' was in job stream:

```
LOC. BYTE 60=031075
```

- 3) / DATE '031075'
/ EXEC DL11
/ PARAM 0001,01,2
/ DATA C
CTL N,W,...ETC
PRGM TEST,* THIS IS A COMMENT

The above linker statements will result in the ELT record's comment field to look as follows:

```
LOC. BYTE 60=031075 THIS IS A COMM
```

Corrections:

1. Made the "flags" feature operative so that auto-inclusion will work correctly with the Rev. 8.0 FORTRAN library, and as documented in the Tape/Disc Assembler manual. If the linker is doing a library search for automatic inclusion of relocatable elements, it will look at an element in the relocatable library only if it
 - (a) has NO flags OR
 - (b) has identical flags to those specified in the PRGM card.If no flags were specified in the PRGM card, elements containing any flags will not be considered for automatic inclusion.
2. Corrected the linker's test for valid L.U. # in a command to test for no greater than 63 (X'3F') instead of 80. Highest l.u.# permissible with our LUPU proc is 63.

NOTE:

At least two blanks must follow the last positional option of the linker (PRGM CHAIN, or SYMB) statement before a comment is given.

APPENDIX K. LIBRARIAN (LIBS)

A. The following enhancements have been made:

1. In a disc-tape LIBS all tape units except those allocated to a symbiont will be accessible for I/O activity in a LIBS run.
2. When "TAB" is specified in a LIBS command, a check is now made on the special comments card (the card has a "." in cols. 1 and 2) of a source module. This card will not be formatted.

B. The following correction has been made:

When relocatable elements are to be changed via the DATA command card, if no ESID is specified in Positional Parameter 2, an ESID of 1 will be assumed.

C. LINKING PROCEDURES:

With this release, six (6) different relocatable elements are supplied, any one of which may be used as the main LIBS program. Users who wish to link more than one version of LIBS may rename the different loadable versions in order to distinguish between them on the same system disc.

The following table indicates the relocatable elements available:

<u>ELTNAME</u>	<u>CALLS</u>
LIBSØ1	DISC=8411,CTL=YES
LIBSØ2	DISC=8411,CTL=YES, PUNCH=YES
LIBSØ3	DISC=8411,CTL=YES,PUNCH=YES, SEQ=YES,TAB=YES
LIBSØ5	as LIBSØ1, plus TAPE=YES
LIBSØ6	as LIBSØ2, plus TAPE=YES
LIBSØ7	as LIBSØ3, plus TAPE=YES

The new LIBS update (UP-8028-A) should be consulted for detailed information on the parameters which may be used to generate the main librarian program.

The linker statements for linking a version of LIBS using LIBSØ2 are provided in the utilities linker deck, CDS. A different version of LIBS may be produced by substituting LIBSxx for LIBSØ2 in this link deck. If LIBSØ5 through LIBSØ7 is included, the following statement must be inserted after "INCLUDE /IO11":

```
INCLUDE /IOTP
```

If LIBSØ1 or LIBSØ5 is included, the "INCLUDE /PCH" and "INCLUDE /TBPU" statements should be removed from the linker deck.

Users may substitute PCW (for row punch) or PCH4 (for 1ØØ4 punch) for PCH when linking a LIBS which will include punched output capabilities. LIBS does not open the punch IOCS routine which is linked to it unless an input statement requests the punching of a module.

NOTE: The IPL equate card is no longer necessary in the linking of the various LIBS versions.

D. SOFTWARE RESTRICTIONS:

1. It is likely that versions of LIBS which include all possible options may not fit into available memory when linked. This is especially true in 16K system, or when a large COS supervisor is generated and the user customarily keeps one or more symbionts resident. The linker map should be consulted to determine the size of the linked librarian.
2. The following are known problems:
 - a. Punching of more than one element with the COPY or CHANGE command results in the loss of ELT records after the first element.
 - b. Sequencing of source code of more than one element with the COPY or CHANGE command results in the resequencing of all modules preceding and including the element.

E. A X'20' will be 'OR'd' onto the UPSI byte by LIBS if an error occurs during its execution.

NOTE: These LIBS elements (LIBS04, LIBS08, and XLIBS) are no longer supplied as of this release. These were elements which included the cross-referencing feature. The cross-referencing feature has been deleted from LIBS. See Appendix Q for the new disc cross-reference utility; and Appendix P for the new tape cross-reference utility.

F. Consult the TAPE DISC LIBRARIAN manual UP-8028-A for new features and updates specified in the command card.

G. ATTENTION:

It is not recommended to use "BOG" records in a library run with the same name.

EXAMPLE:

▽	BOG	A
▽	BOG	B
▽	BOG	C
▽	ELT	ONE
▽	EOG	
▽	EOG	
▽	EOG	
* ▽	BOG	A
* ▽	BOG	B
* ▽	BOG	C
▽	ELT	TWO
▽	EOG	
▽	EOG	
▽	EOG	
/*		

* It is suggested you do not use the same name for two (or more) different groups.

APPENDIX L . MISCELLANEOUS IOCS.

The corrections and improvements described below have been made to the IOCS routines indicated therein.

1. Control Stream Reader (DTFCS).

- a. The HSRB parameter has been eliminated.
- b. Corrected failure to issue 41FD and 41FE messages as expecting a reply; this made recovery from these messages impossible in OS-500.
- c. Corrected failure of DTFCS to initialize the control stream when the program being executed was loaded through the control stream reader (/ * in columns 16-17 of EXEC card); this led to unnecessary 41FE and 41CF stops.

2. Serial Read/Punch (DTFRP).

Corrected undefined flags caused when generating DTFRP for combined files.

3. 1004 Printer (DTF4P).

Corrected 1004 printer's IOCS hanging the system when form overflow occurred and DTF4P was generated with CNTL=YES and PROV omitted.

4. Printer/Control Stream Transient (RD??PRNT).

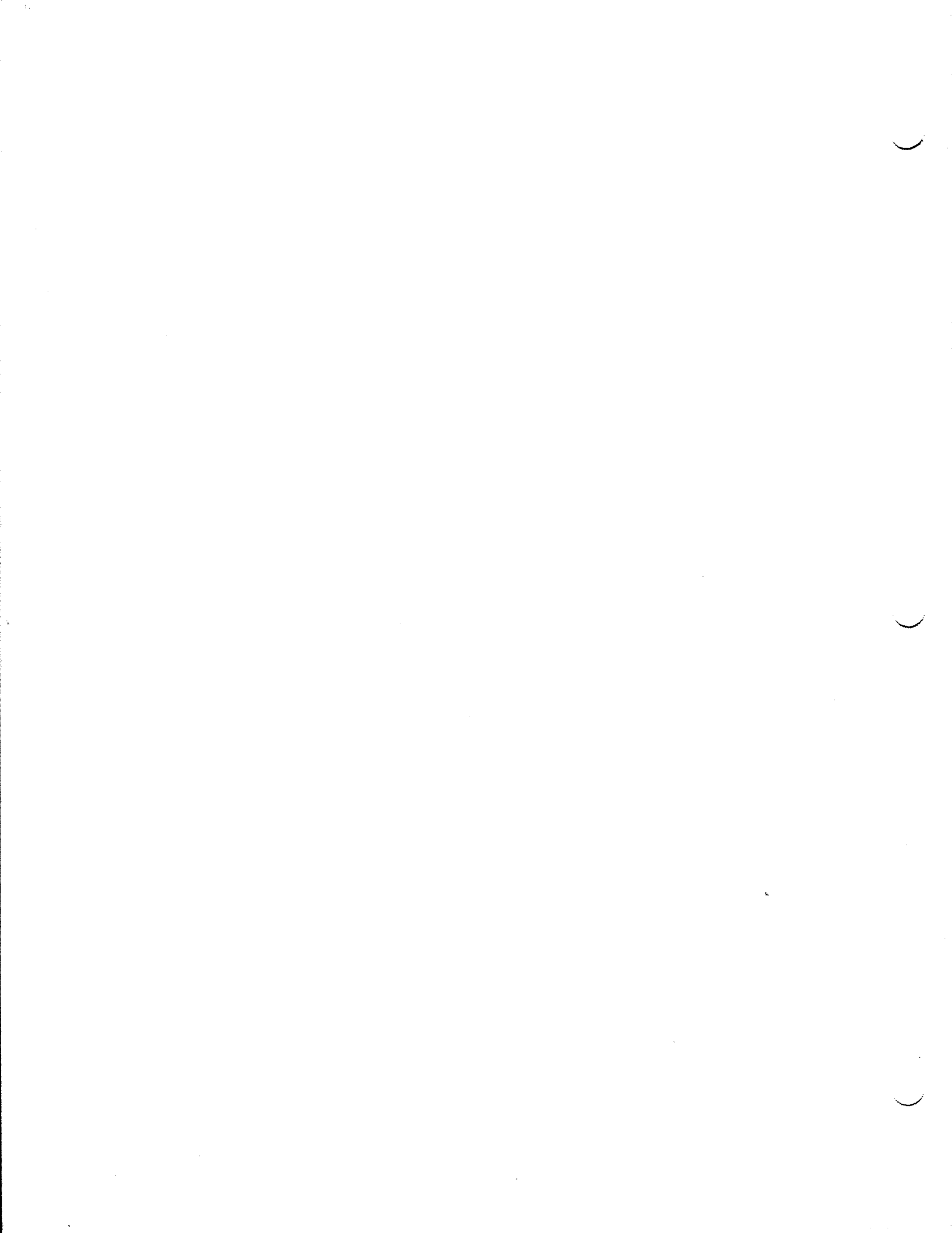
This transient's handling of printer delayed error recovery procedures in an OS-500 environment is now identical with that of DTFPR. The printer is now always restarted with a X'D3' keyin under OS-500.

5. Row Punch (DTFRW).

- a. When the row punch IOCS issued a 64Ø1 or 64ØF MSG, a reply was not required; in an OS-500 environment, this caused the system to loop, printing out the same MSG. A reply is now required to each of these displays.
- b. Corrected the possibility of PROC ABN occurring if the row punch IOCS was opened and closed without any intervening PUT taking place.

6. When generating the Disc-to-Bar printer symbiont (with parameter BAR=YES) and parameter CHNL=n and SYMB=n are omitted, they received the values CHNL=Ø and SYMB=Ø. Corrected problem to have a default of CHNL=3 and SYMB=3.

When generating the Disc-to-Drum printer symbiont the CHNL=n must be specified. The SYMB has a default value of 3.



APPENDIX M. OS-500

This Appendix describes features of this software release that are of interest only to OS-500 users. In OS-500 areas, few bugs remained to be corrected and emphasis has therefore been given to making the software more efficient and easier to use. While it will probably be necessary for you to reassemble many of your inquiry programs to operate under DNCOS 5.0, we believe that the benefits realized from upgrading will more than compensate for the time spent reassembling.

1. New Supervisor Generation Options.

- a. ROLL - To generate a system which is capable of rolling out the main ("batch") job to process inquiries, ROLL=YES is now specified. Roll-out always takes place to the system resident unit; the ROLU parameter has been deleted. If the ROLL parameter is omitted, the roll-out capability is not generated.
- b. STATE - Formerly, the main job could only be suspended during the execution of a program for which a "PARAM S" card was used or during the Job Control FINIS loop. You now have the option of defining, at system generation time, whether the normal state of your system is to be suspendible or non-suspendible. If STATE=NOSUSP is specified, the suspendibility of the system is as before. If STATE=SUSP is specified, or if the STATE parameter is omitted, the system is always suspendible except when a program is being executed for which a "PARAM N" card is submitted.
- c. INQPL - When a request was received to activate a program which was not resident, the requested program was searched for in SYSFILE on the system resident unit. This was often inconvenient because SYSFILE would contain numerous other elements and in any event was slow because of the many disc accesses required for locating and loading programs from library files. For these reasons, we have implemented the System Execution Area (SYSEXAR) specifically for loading of inquiry or other often-used programs. If the OS-500 Supervisor is generated with ROLL=YES specified, inquiry programs will be "rolled in" by the fast loader unless INQPL=BOTH or INQPL=SLOW is specified. The former implies that inquiry programs are to be searched for first in SYSEXAR by the fast loader, then in SYSFILE by the slow loader. The specification of INQPL=SLOW keeps the loading of non-resident inquiry programs as is, i.e. from SYSFILE. We strongly recommend use of the fast loader (INQPL omitted or INQPL=FAST) and have included the other options only to facilitate conversion. Response time has been found to decrease from 5-10 seconds to 1-3 seconds. Of course, the amount of time the batch job is suspended is also dependent on the processing and disc accesses performed by the inquiry program once it is loaded.

2. Roll-in/Roll-out.

As noted in 1a. above, roll-out is now done on the system resident unit. We believe that this is safer than specifying a logical unit since any swaps which occurred during the course of a run would cause roll-out to a different disc pack with possibly disastrous results. The following additional changes have been made to OS-500's roll-out philosophy:

- a. The main job is no longer rolled out to an arbitrarily-chosen cylinder, but to a file whose i.d. is "ROLLOUT". This file may be located wherever desired by the user, but we recommend keeping it close to TRANSCYL, SYSEXAR and SYSFILE in order to keep arm movement to a minimum. ROLLOUT must be cylinder-aligned. The first track of ROLLOUT is used for rolling out the transient area and miscellaneous Supervisor tables. Enough additional tracks must be allocated to ROLLOUT to permit rolling out the entire area from beginning of physical unit tables to the top of memory. (In a concurrent system, the area occupied by symbionts is not rolled out, but the disc space must exist in case no symbionts are present when an inquiry is received.)
- b. ROLLOUT is now pre-formatted by Job Control each time the system is booted. This is done to permit faster roll-out and roll-in. Job Control also ensures that sufficient tracks were allocated to the file to permit rolling out the largest possible main job. Job Control is aware of whether the system was booted from 8411 or 8414 and takes advantage of the larger 8414 track capacity; formerly OS-500 rolled out only 3625 bytes per track.
- c. If no file named ROLLOUT has been allocated and a request is received to activate a non-resident program, the inquiring station receives an INQUIRY-ABORTED response.

3. Inquiry Program Activation.

- a. The former requirement of typing "//RUN,", plus a 7-character program-i.d., was found to be slow and cumbersome. Those who wish to do so may still type "//RUN," preceding the desired program-i.d., but its use is now optional. In an OS-500 system that has inquiry capability any unsolicited type-in that does not begin with a slash is taken to be a request to activate a program. The program-i.d. may consist of from one to eight characters. If the program to be activated is a single-inquiry mode program, the program-i.d. must be followed by a comma which in turn is followed by the input data.

Example:

PERS, JOHN JONES

would activate a program named PERS, with "JOHN JONES" being the input data to that program.

If the OS-500 monitor scans more than eight characters without encountering either a blank or comma, a REJECT-INPUT message is sent. The i.d. typed in must match, character-for-character, the name used when linking or compiling the program, i.e. "old" programs which had M or S appended to their names can be activated by the new Supervisor only if all eight characters are typed in.

- b. When an inquiry was received for a non-resident program, the OS-500 monitor formerly waited until the next SRC issued by the main job to try to roll out the main job. It now tries to roll out the main job immediately. If the Proc PSC is within the Supervisor but not within the system transient area or in the control stream reader get-next-card wait loop, roll-out is deferred until the next interrupt from any source at which time the address within the Proc PSC is once again checked.
- c. Problems have been corrected within the Supervisor concerning the use of the single inquiry work area for system type-ins (/KYN, /ANS, etc.) while an inquiry program was active. Any system type-in, except for a request to activate an inquiry program, can now be processed while an inquiry program is active. An inquiry from the system console while another inquiry is active is rejected with a "REJECT-INPUT" response.

9200/9200II/9300/9300II
Memo #147

4. Control Stream Logging.

Various inconsistencies in Job Control's logging of the control stream have been corrected. The following will explain current practice:

- a. Each statement which is read by Job Control is logged only after it has been identified. If any error is detected in its contents, the MSG caused by the error will be printed after the control card image.
- b. If CLOG=JECF is specified, in addition to JOB, EXEC, CALL and FINIS statements, RUN and NOTE statements are also logged.
- c. If CLOG=ALL is specified, Job Control sets the X'40' bit at E\$FLAGS in the Supervisor. This bit is tested by the PARAM card processing transient (PARAM?); if it is on, each PARAM card is logged, as well as the DATA READER statement if the system includes the SJCS option. Again, each card is logged only after it has been identified.

If either CLOG=ALL or CLOG=JECF is specified, Job Control prints a message upon initial load that now includes the logical unit number, device address, and volume serial number of the system resident disc unit.

5. OPR Macro Instruction Implementation.

a. Programming Considerations.

The format of the OPR Macro Instruction is:

<u>LABEL:</u>	<u>OPERATION:</u>	<u>OPERAND(S)</u>
[tag]	OPR	addr [mm] [x] [nn]

WHERE:

- tag is the optional symbol to be attributed to the line containing the OPR.
- addr is the symbolic location of data which is to be displayed upon the system console.
- mm is the length of the data to be displayed. Maximum permissible value is 255. 64 is assumed if mm is omitted.
- x is any character; this parameter is ignored by the Assembler and is present only for compatibility with the format of the OPR macro under OS-4.
- nn is the maximum length of the expected reply. If nn is omitted, this signifies that no reply is expected. Maximum permissible value is 255.

addr, mm, and nn may be any valid Assembler expression. The machine code produced by the Assembler is as follows:

```
DC      Y(X'A108')          SRC opcode; same as MSG
DC      Y(addr)
DC      Y(mm)
DC      X'40'
DC      YL1(nn)
```

The OPR Macro Instruction is an "internal" Assembler macro, just as are OPEN, CLOSE, MSG, FETCH, etc. It may be assembled by any new tape or disc assembler. It can only be executed meaningfully under the control of an OS-500 Supervisor. The OPR feature is available regardless of whether the Supervisor includes inquiry capabilities.

Examples:

- (1) OPR AREA1
- (2) OPR AREA2,25
- (3) MVC OPRWORK(12),=C'TEST MESSAGE'
OPR OPRWORK,12,,1
- (4) MVC OPRWORK(L'MSG1),MSG1
OPR OPRWORK,L'MSG1,X,L'MSG1
- (5) OPR TRASH,5,,5

- (1) The contents of an area labelled AREA1 are to be displayed; the assembler inserts the default length of 64 bytes.
- (2) The 25 bytes beginning at AREA2 are to be displayed.
- (3) "TEST MESSAGE" is to be displayed; a non-byte reply is expected and will be available in OPRWORK. Note that the third operand ("x") must be accounted for by a comma even though it is omitted.
- (4) The contents of MSG1 are moved to OPRWORK and displayed; a reply is expected whose length is to be no longer than that of the data displayed.
- (5) The contents of the area labelled TRASH are to be displayed and are to be overlaid by the operator's reply.

b. Operating Considerations.

The Supervisor distinguishes between an OPR and an MSG by examining bytes 2 and 3 of the SRC instruction; if they are non-zero, OPR is assumed. OPR always causes output to, and input from, the system console. If the output area contains trailing blanks, the OS-500 monitor scans to the last significant character and only causes the number of significant characters to be written on the console. This maximizes throughput and permits the programmer to use one work area to put out messages of various lengths without having to concern himself with the length of each possible message. The length of the work area itself may be specified in the OPR as long as the area is cleared to blanks before moving output data to it. If a reply is expected, OS-500 clears the message area to blanks before turning on the clear-to-send light. Since OPR uses much of the previously-existing MSG code in the Supervisor, its impact on Supervisor size is minimal. There is no restriction on the issuance of OPR macros by symbionts. The program or symbiont issuing the OPR is suspended until the data to be displayed is typed out and the reply (if desired) is received.

Since OPR is similar in use to MSG, and since roll-out cannot take place with either an MSG or OPR outstanding, programmers should use discretion when writing OPR's in programs in an environment where many inquiries are taking place. OPR may be used by inquiry programs to display messages on the system console without having to resort to CNTRL macros to change DTFUQ's inquiry station's logical unit number.

The Supervisor issues a carriage return/line-feed function before printing the OPR data. If a reply is desired, another carriage return/line-feed is issued before accepting the reply.

6. Suspendibility.

In keeping with the new concept of suspendibility described in 1b. above, we no longer arbitrarily make the system non-suspendible during the functioning of Job Control. The suspendibility bit of the monitor activity byte is set or reset at only three times:

- a. It is always set when Job Control enters its FINIS loop.
- b. It is always set by the Inquiry IOCS routine (generated by DTFUQ declarative macro) of an inquiry program at initialization (IN?U). This permits the roll-out of any inquiry program which is loaded through the control stream while it is not actually processing an inquiry.
- c. When any main program is loaded, after all PARAM cards have been processed, the system's state of suspendibility is set/reset in accordance with the STATE option chosen at system generation time unless this "normal" state is overridden by a PARAM N or PARAM S card. The former card is recognized only if the system's normal state is suspendible; the latter only if its normal state is nonsuspendible. Once set at program-load time, the system's state of suspendibility remains the same through the subsequent job step processing by Job Control up to the point where the next program is about to be given control.

7. Use of the SOE Character.

Changes have been made to the Supervisor and to DTFUQ to permit the omission of the SOE character in input from a Uniscope. It is no longer assumed that all input data starts with a SOE. When activating a program or scanning data for an inquiry program, the SOE is tested for but ignored. On output to a Uniscope for printing under program control, a SOE is inserted only when the output message begins at other than the first row and column of the screen.

8. Suggestions for Testing and Debugging Inquiry Programs.

The techniques described herein are based on discussion of inquiry programming problems and methods with many experienced programmers. Although each application may seem unique, there are certain methods of testing new programs which may simplify the programmer's tasks.

a. Local Console Use.

When testing an inquiry program, it is to the programmer's advantage to be in the general vicinity of the computer. The need to maintain parallel communications between the on-site operator and the programmer at a remote Uniscope or DCT-500 (or vice versa) only adds to the problems inherent in any debugging situation. Even when the "remote" inquiry device is in the same room, the uncertainties of modems, DCS, and line conditions often add confusion to the picture. We therefore suggest that all inquiry programs be tested initially through the system console. This permits the programmer to eliminate bugs in his own processing logic as quickly as possible. Even if the program, when placed into production, will function only from a remote, those features of it which are dependent on remotes (scope coordinates or CNTRL macros to printers or paper tape units) can be bypassed until the basic user program logic is functional.

b. Execution Through Control Stream.

When a new program is being debugged, it is invariably necessary to make on-the-spot corrections to insert halts, bypass inoperative code, and otherwise modify the program without having to reassemble/recompile/relink. Patching is difficult or impossible if the program is being "rolled in" from SYSEXAR or SYSFILE. It may also be bothersome to carry out LIBS or SHEAF updates each time the program is regenerated. For these reasons, we suggest that debugging of new, or updated, inquiry programs take place by executing the program through the control stream from whatever work file it was linked into. This facilitates the use of PARAM cards and manual keyins for quick corrections.

c. DTFUQ Generation Listing.

It is difficult to debug a new program without seeing what is taking place inside the Inquiry IOCS routine generated by DTFUQ. We have made every effort to document DTFUQ's operation by the insertion of meaningful comments. The PRINT NOGEN statement should not be used, at least for the first time a new inquiry program is being assembled. If your UNIVAC Systems Analyst must communicate with Blue Bell concerning problems in inquiry programs, it will be necessary for him to have a generated listing of the DTFUQ used by your program available for comparison and possibly for making fixes to problem areas. If you wish to keep reassembly time to a minimum, do not assemble DTFUQ with each worker program, but keep a "standard" set of DTFUQ-generated relocatable elements in a permanent library from which they may be linked to any inquiry program.

d. Non-Transient DTFUQ.

It is sometimes necessary to insert stops within DTFUQ in order to trap a particularly bothersome problem, even though the problem may not be caused by software failures. When memory space permits, while an inquiry program is being debugged, its Inquiry IOCS should be one that was generated without TRAN=YES being specified. This permits you to see the entire IOCS routine being used by your program and to isolate and correct more quickly whatever problems may arise. If available memory is limited, make the heaviest possible use of Disc IOCS transients while debugging to compensate for using a non-transient DTFUQ.

9. Corrections and Improvements in DTFUQ.

- a. Many corrections were made to DTFUQ to permit it to interface correctly with changes made to the Supervisor. The following restrictions must be made on the use of programs generated by previous levels of DTFUQ with the new Supervisor:
 - (1) Any program whose DTFUQ was generated with TRAN=YES specified is no longer operational and must be relinked/regenerated with a new DTFUQ.
 - (2) Any program which operated in the single inquiry mode (SINQ=YES) can be activated under the new Supervisor only by typing in "//RUN," followed by all eight characters of its program-i.d.
- b. The NACL parameter and all code associated with it has been deleted from DTFUQ. Its use was made unnecessary by the implementation of the OPR macro.
- c. The processing of input carriage returns was different if the input was from a Uniscope or from a local console or DCT-500. Carriage return characters (X'0D') are now eliminated from all input unless IPCR=YES is specified.
- d. DTFUQ's generation of diagnostics for possibly erroneous combinations of parameters has been improved. Wherever possible, default assumptions are made; they are described by MNOTEs. If errors are detected which would prevent the generation of valid code, they are flagged by PNOTEs and the generation of DTFUQ is aborted.
- e. The voluminous comment lines which formerly were included in the DTFUQ Proc have been eliminated; all information pertinent to the generation and use of DTFUQ is contained in the OS-500 Programmers Reference Manual, as updated and amended by this release memo.
- f. Various errors in DTFUQ's data translation and move routines have been corrected. These errors formerly caused garbage to be transmitted, abort status, and incorrect lengths to be inserted in input data.
- g. It is no longer necessary that the work area defined in the GET or PUT instruction be halfword-aligned.
- h. The inquiry IOCS routine now checks, in all generations, for the file being opened. Issuance of OPEN to a previously opened file or of other macros to a file that is not open will cause a X'4171' halt.
- i. The inquiry IOCS now makes only one scan through the data being transmitted or received; it formerly made as many as three. We believe this will increase throughput significantly in programs which transmit and receive entire screens of data. Output to a local station is done without any scan being made for purposes of shifting or translating.

- j. Due to the many changes being made, it was necessary to break the Inquiry IOCS Transient down into two transients, one for OPEN and CLOSE processing and the other for GET/PUT/CNTRL. Since the two new transients are each considerably smaller than the former transient, this change has permitted us to cut the size of the system transient area and thereby reduce slightly the size of the resident Supervisor.
- k. Due to the design of the Inquiry IOCS transients, the following restriction must be observed if TRAN=YES is specified:

If the CNTRL macro is issued for any purpose, no processing which could cause the loading of any other transient can intervene between the CNTRL and the succeeding GET or PUT macro.

- 10. In addition to the changes described above, the following information from preceding release memos is still applicable:

Corrections to UP-7869 (Rev. 3):

- a. Page 2-11 - The last sentence under "Error" should read:

When the UNISCOPE 100 is the remote device, ERRO=label must be specified. ERRO=IGNORE is not an acceptable specification.

- b. Page 2-13 - The explanation of U100=YES should read:

YES specifies that this generation of DTFUQ will be used for processing inquiries from remote UNISCOPE 100's.

- c. Page 3-1 - The last sentence of paragraph 3.2 should read:

When typing in messages to the system or input data to a user program which uses DTFUQ, the operator does not need to hold down the shift key to type in alphabetic characters. If a user program issues its own requests to the OS-500 monitor (SRC 0,52), it must either require that the operator hold down the shift key to type in alphabetic characters, or else OR on the X'40' bit of each alphabetic character received.

- d. Pages 3-3 and 3-4 - The following information supplements or corrects the information provided in the manual:

- (1) OS-500 differentiates between console displays ("//MSG" type-outs) which require answers and those which do not. The issuance of the former by a main program or symbiont will cause that program to be suspended until such time as the operator replies to the message by a "//ANS" type-in. The issuance of the latter will cause control to be returned to the issuing program as soon as the display has been typed out on the console typewriter. The operator may distinguish between the two types of messages by the presence or absence of the letter "A" (signifying "answer required") following the four-digit message. Only those displays which are caused by a MSG macro which includes the "REPLY" parameter will result in an answer-required situation. Programs which execute message macros for the purpose of causing operator action such as changing printer forms, correcting erroneous input cards, etc., should issue a MSG with REPLY specified, even though the value of such reply is of no interest to the program. All MSG macros in the Supervisor and Job Control program include the reply parameter since they usually require that the operator take some action (changing tapes or disc packs, correcting job control cards) before processing can continue. MSG macros in other software elements have been examined for the desirability of requiring a reply and have been converted on a selective basis. Any routine which operates in I/O mode and issues console messages via a branch-and-link to E?DS must not expect a reply to be available to it. Control will be returned to it as soon as the message has been typed out.

- (2) When the main job is cancelled or a symbiont is killed, OS-500 clears the message table of any pertinent messages, thus preventing the wiping out of another program if the operator attempts to answer a "dead" message.
- (3) Since messages issued by routines operating in I/O mode do not require a reply (see "1." above), when the operator wishes to restart a device which was placed in error status, he simply types in "//KYN,Dx", with x being device type. See UP-7531 (Rev. 5), page 3-6, for a table of acceptable device types.
- (4) Disc and tape error recovery procedures conform to error recovery procedures for other device types. When a tape or disc unit is in error status, it may be restarted by typing in:

```
//KYN,Dx,mmmm,rr      OR
//KYN,Dx,mmmm
```

"x" is \emptyset for tapes, 7 for discs; "mmmm" is the message which announced the drive's error status to the operator; "rr" is the one-byte reply which the halt display manual requires for the particular message. If "rr" is omitted, zero is assumed; in most cases, the zero reply signifies a retry attempt.

- (5) Under OS-500, no MSG may have a value greater than X'7FFF'.

e. Page 4-18:

- (1) When generating any disc-resident supervisor, the symbol "E?XS" must appear in the operand field of the "END" statement.
- (2) When generating an OS-500 that includes remote inquiry stations, COMM=YES must be specified in supervisor generation.

f. Page 4-19:

The opcode on the first line of the coding sheet should read "CFG \emptyset A".

g. Page 4-21:

- (1) When generating Job Control for an OS-500 system, OS5H=YES must be specified. If the system includes remote devices, COMM=YES must also be specified.
- (2) When generating any Job Control program, the symbol "STRT" must appear in the operand field of the "END" statement.
- (3) When generating Job Control for an OS-500 system, either CLOG=ALL or CLOG=JECF must be specified.

11. System Transient Area Size.

The size of the system transient area for OS-500 is X' \emptyset 52 \emptyset ' (1312 \emptyset) bytes except when the Supervisor is generated as follows:

ROLL=YES plus INQPL=SLOW or INQPL=BOTH, size is X' \emptyset 56 \emptyset ' (1376 \emptyset) bytes.

same, plus concurrency (OS5C), size is X' \emptyset 5A \emptyset ' (144 \emptyset) bytes.

APPENDIX N . 8411/8414 DISC UTILITIES (DASM, DPRT, ISF)

A. DASM

Problems Corrected

1. When DASM detected an error in the parameter cards of an 8414 DASM run, it inadvertently reset itself to DASM an 8411 pack.
2. The disc 'track size calculation' routine was in error on various block sizes in its calculation of blocks per track for both 8411 and 8414.

Enhancement

DASM has been modified to permit an optional 'END' control card. If a missing 'END' card is detected in a control stream, a pseudo end card image '.END' will be printed within the VTOC input diagnostics section and processing will be continued with a dummy, map-option 'END' card. In addition, this enhancement eliminates the X'12FF' halt.

For example, the following two control streams will obtain identical results.

DVC	DVC
VOL	VOL
LBL	LBL
EXT	EXT
DVC	END*
VOL	DVC
LBL	VOL
EXT	LBL
	EXT
	END*

*Note: The END control card is not compulsory.

B. DPRT

Problems Corrected

1. Valid logical unit numbers specified larger than X'09' would not print the logical unit # correctly, although the correct logical unit was processed.
2. A retry response to a X'69u2' display would result in a loop.

C. ISF

Problems Corrected

1. No check was made to prove that block size (NREC*RCSZ) plus key length (KLEN) was not greater than track size for the applicable disc (8411/8414). If specifications are made where BKSZ+KLEN is greater than 3605 for 8411's (7249 for 8414's) an error message ***BKSX+KLEN>TRACK*** will be printed and the ISF run will be cancelled.

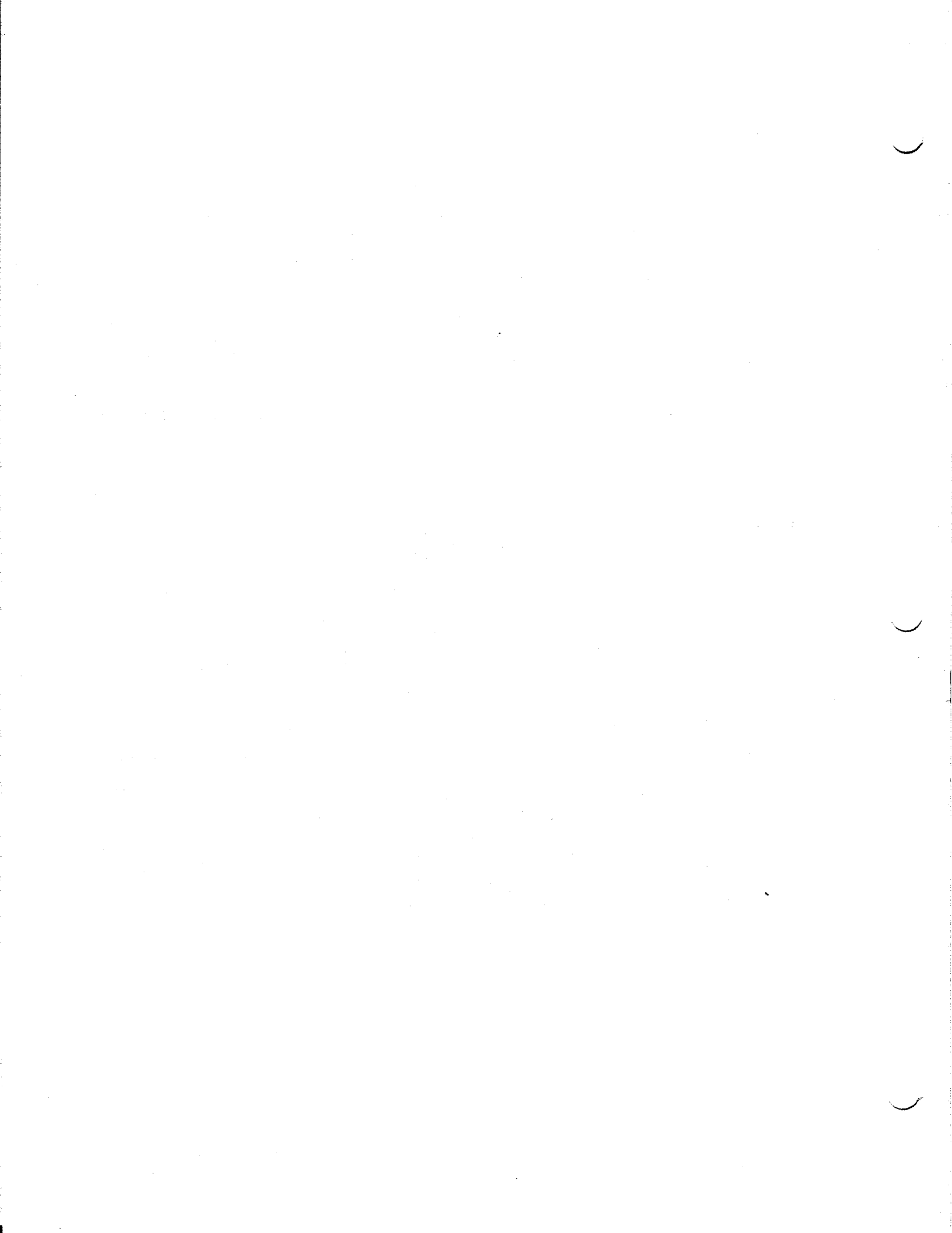
2. Cylinder Index extent alignment was incorrect when the cylinder index did not begin on a cylinder boundary.
3. No check was made to prove that the space required for the cylinder index (as calculated by ISF) was not larger than the space specified as allocated by the cylinder extent. If this does occur, an error message *****INSF INDEX TRACKS***** will be printed and the ISF run will be cancelled.

APPENDIX O. DUMP/RESTORE UTILITY (DPRS)

1. The following DPRS bugs have been corrected:
 - a. When restoring a multi-reel dump tape, DPRS would reject the second reel of the tape as having the wrong file header.
 - b. When dumping a file which contained an end-of-file mark at the beginning of a track (record one), the end-of-file mark was not copied.
2. DPRS execution times have been greatly improved by changing its logic to read a track at a time instead of a record at a time. The following timings are for dumping/restoring entire packs and should indicate the relative increase in throughput:

Function:	<u>8411-to-8411</u>	<u>8414-to-8414</u>	<u>8411-to-tape</u>	<u>8414-to-tape</u>	<u>tape-to-8411</u>	<u>tape-to-8414</u>
Old time:	28	74	29	65	19	19
New Time:	7	8	6	8	6	9

3. To implement this increase in speed, it was necessary to simplify some portions of DPRS. The following restrictions now exist:
 - a. DPRS must have enough room in memory to hold an entire 8414 track. This requires that if it is to be used in a 16-K machine, it must be linked with PCH equated to \emptyset and no punch IOCS included. If DPRS finds that it does not have room to hold an 8414 track, it will try to make more room available for a disc-to-disc dump by overlaying its tape and card dump capabilities. If it does so, it will print self-explanatory messages to that effect on the bar printer.
 - b. Dump tapes which contain "partial-track" dumps (X'CC' in the last byte) can no longer be restored. To determine whether a tape might contain partial track blocks, consult the printer listing produced when the tape was created. If the "available work area size" was less than track capacity, it is possible that partial track blocks exist.
 - c. Display 2Bh3 is no longer given. If no punch IOCS was linked to DPRS or if it was necessary to delete card I/O capabilities to make room for disc-to-disc dumping, display 2Bh1 now occurs.
 - d. Since all dumping and restoring is done on a track-at-a-time basis, all error printouts which result from read/write errors will give only cylinder and head addresses. If any read/write error occurs for a track, one dummy block will be generated for the track if the operator's reply is to continue processing.



APPENDIX P. TAPE CROSS-REFERENCE UTILITY

A. A new tape cross-referencing utility is now available.

The only function of this utility is to cross-reference assembler source code from either tape or the control stream reader.

NOTE: The tape cross-referencer can be used in a disc environment only if the generated system has tape capabilities.

Input is controlled by command cards. The control character (specified when linking by "DEL") must appear in column 1 followed by at least one blank character. The "command" may be any word beginning with an alphabetic character.

1. CARD INPUT. The command card has the following format:

∇ COMMAND CARD\$

EXAMPLE:

∇ COPY CARD\$

To list a module while cross-referencing card input, the "command" word must begin with the letter "L".

EXAMPLE:

∇ LIST CARD\$

Listings will respond to SPACE, EJECT, and TITLE directives within the source element.

CARD\$ identifies the input as coming from the control stream reader. The source deck must immediately follow the "command" card.

2. TAPE INPUT. The "command" card has the following format:

∇ COMMAND GROUP-NAME/MODULE-NAME,L.U.#

The "command" may be any word beginning with an alphabetic character. This could eliminate keypunching a new card if the card has been used in a LIBS run.

EXAMPLE:

∇ COPY /XREF

"GROUP-NAME" must be in the same format as a library services command. "MODULE-NAME" is the name of the source code module to be cross-referenced. "L.U.#" is the logical unit number on which the module resides, expressed as one or two valid hexadecimal digits.

EXAMPLE:

∇ COPY MACROLIB/UNIVAC/EXECMODL/JBCN,00

NOTE: If the logical unit number has not yet been specified, the default is to logical unit 0, otherwise the logical unit number from the previous "command" card will be used.

3. No further cards will be read from the control stream until the particular module in the "command" card has been cross-referenced.

Macro definitions may be included in the source code. All symbolic parameters containing an ampersand or a percent sign will be ignored. Keyword parameters will not be cross-referenced (e.g. TAPE=YES).

Any number of modules, in any combination of source inputs, may be cross-referenced in one run.

Each "command" card is checked for validity and invalid cards are flagged and ignored.

All literals and constants in the operand portion of the source input record will be ignored, except for address constants.

EXAMPLE 1:

```
CLC      TAGA,=CL2Ø' '
```

where =CL2Ø' ' will not be cross-referenced.

EXAMPLE 2:

```
CLC      TAGA,=Y(TAGB)
```

where TAGB will be cross-referenced.

NOTE: No output tape is required.

NODATE will be printed on every header page of the listing if the supervisor's date contains all 1 bits (6 bytes of X'FF').

A 'DATE' card placed in front of the run deck will override the 'NO DATE' printout.

B. LINKING PROCEDURES:

The following elements are required to link the cross-reference utility:

1. TXREF - the relocatable supplied - for TAPE and CARD cross-referencing.
2. TPRYTR- the relocatable transient supplied for the printer I/O. This deck is for the bar printer only. If the user desired to output his listing on the drum printer or the 1ØØ4 printer, etc., the user must assemble a new DTF element which uses such printers. The filename of this IOCS routine should be 'TPRY', and the printer overflow routine must be specified as PROV=FFOV.
3. PLUN - the relocatable logical unit tables element generated during SYSGEN procedures.
4. DEL EQU 'n' - where 'n' (as in the LIBS manual) is the compressed code value, expressed in decimal or hexadecimal notation, of the control punches in column 1 of the "command" cards. For example, if √ is desired as the control character, DEL EQU 85 or DEL EQU X'55' must be specified, and the punch is 12-Ø-2.

NOTE: A UNITS statement must be used when linking the tape cross-referencer if a transient TAPE dispatcher is desired. Refer to Operating System Manual, UP-7531 Rev. 5, Sections 8.6 through 8.6.6.

It is possible to use the same card as in a LIBS run deck to cross-reference a module by equating DEL to the same value as when LIBS was linked.

It is not necessary to link any control stream reader element to the program, since it is part of the main cross-reference element - TXREF.

Cards are read from the control stream read device as specified at systems generation time.

When linking the cross-referencer, TXREF must always appear last.

EXAMPLE: (a control stream linker deck run)

```
/      EXEC    DL11
/      DATA   C
      CTL     card
      PRGM    TREF,*
      LIBE    NONE
DEL    EQU     X'55'      (or DEL EQU 85)
      INCLUDE /TPRYTR
      INCLUDE /PLUN
      INCLUDE /TXREF
/*
```

With this new routine it is now possible to cross-reference source code in 16K because of a recycle feature. Cross-referencing in a 16K tape environment was a previous restriction with LIBS.

Symbols, when cross-referenced, will appear in alphabetical order on the printout.

The cross-referencer uses all memory from the end of where the program (TXREF) is linked to the end of available memory for storage of symbols, their references, etc.

The contents of the boundary table will be adjusted to reflect the memory available to the cross-reference routine when Symbionts are used.

It is advisable to run the cross-referencer without symbionts because of memory limitation to prevent the cross-referencer from recycling and making more passes of the input source code.

C. RESTRICTIONS:

At this time, it is impossible to cross-reference a group. Each element to be cross-referenced must be specified.

D. ERROR HANDLING:

See chart in Appendix Q , paragraph D.

E. SPECIAL NOTE: RECYCLING

When the symbol table has filled all available memory, recycling occurs if more symbols are to be processed. As a result, certain symbols may have "***NODEF**", or "***UNREFERENCED***" printed beside them. These symbols should be checked since their definitions and/or references may appear on subsequent or previous recycle passes.

F. HARDWARE CONFIGURATION:

The 9200/II Configuration must be equipped with the hardware multiply/divide/edit features.

G. TAPE PROCESSING CONSIDERATION:

It is advisable to have the "command" cards in the control stream deck in the same sequential order as the modules would appear on tape. This is to avoid getting "MODULE NOT ON TAPE" diagnostics.

The tape will be rewound by the cross-referencer when either of the two following conditions have been met:

1. The 'END' record on tape has been encountered subsequent to cross-referencing the module designated in the "command" card without any intervening 'ELT' record. Processing resumes after the tape is rewound to load point.
2. If after reading another "command" from the control stream and the following conditions are met:
 - a. the tape is not on load point and
 - b. at least one 'ELT' record is detected subsequent to the module that has already been cross-referenced, and
 - c. the module name specified in the current "command" card is not found prior to encountering an 'END' record on that tape.

NOTE: "MODULE NOT ON TAPE" diagnostic will be issued. The next "command" card is processed.

H. DOCUMENTATION:

Changes will be made to the TAPE UTILITY Manual UP-7727 to reflect this new feature.

APPENDIX Q. DISC CROSS-REFERENCING UTILITY

A. A new disc cross-referencing utility is now available.

The only function of this utility is to cross-reference assembler source code from either 8411/14 disc or the control stream reader.

Input is controlled by command cards. The control character (specified when linking by "DEL") must appear in column 1 followed by at least one blank character. The "command" may be any word beginning with an alphabetic character.

1. CARD INPUT. The command card has the following format:

▽ COMMAND CARD\$

EXAMPLE:

▽ COPY CARD\$

To list a module while cross-referencing card input, the "command" word must begin with the letter "L".

EXAMPLE:

▽ LIST CARD\$

Listings will respond to SPACE, EJECT, and TITLE directives within the source element.

CARD\$ identifies the input as coming from the control stream reader. The source deck must immediately follow the "command" card.

2. DISC INPUT. The "command" card has the following format:

▽ COMMAND GROUP-NAME/MODULE-NAME,L.U.#,FILE-NAME

The "command" may be any word beginning with an alphabetic character. This could eliminate key-punching a new card if the card has been used in a LIBS run.

EXAMPLE:

▽ COPY /XREF

If the "command" word, as with card input, begins with the letter "L", the source element from disc will be listed.

EXAMPLE:

▽ LIST /XREF

"GROUP-NAME" must be in the same format as a library services command. "MODULE-NAME" is the name of the source code module to be cross-referenced. "L.U.#" is the logical unit number on which the module resides, expressed as one or two valid hexadecimal digits. "FILE-NAME" is the name of the disc file-id (result of DASM) in which the module is to be found.

EXAMPLE:

▽ COPY MACROLIB/UNIVAC/EXECMODL/JBCN,00,SYSFILE

NOTE: If the logical unit number and file-name has not yet been specified, the default is to logical unit 0, and "SYSFILE". Otherwise the logical unit number and file-name from the previous "command" card will be used.

3. No further cards will be read from the control stream until the particular module in the "command" card has been cross-referenced.

Macro definitions may be included in the source code. All symbolic parameters containing an ampersand or a percent sign will be ignored. Keyword parameters will not be cross-referenced (e.g. DISC=YES).

Any number of modules, in any combination of source inputs, may be cross-referenced in one run.

Each "command" card is checked for validity and invalid cards are flagged and ignored.

All literals and constants in the operand portion of the source input record will be ignored, except for address constants.

EXAMPLE 1:

```
CLC TAGA,=CL20' '
```

where =CL20' ' will not be cross-referenced.

EXAMPLE 2:

```
CLC TAGA,=Y(TAGB)
```

where TAGB will be cross-referenced.

NOTE: No output disc file is required.

NO DATE will be printed on every header page of the listing if the supervisor's date contains all 1 bits (6 bytes of X'FF').

A 'DATE' card placed in front of the run deck will override the 'NO DATE' printout.

B. LINKING PROCEDURES:

The following elements are required to link the cross-reference utility:

1. DXREF - the relocatable supplied - for DISC and CARD cross-referencing.

NOTE: DXREF uses the transient "FILE?" to locate both the file-name and the module-name.

2. TPRYTR- The relocatable transient supplied for the printer I/O. This deck is for the bar printer only. If the user desires to output his listing on the drum printer or the 1004 printer, etc., the user must assemble a new DTF element which uses such printers. The filename of this IOCS routine should be 'TPRY' and the printer overflow routine must be specified as PROV=FFOV.
3. PLUN - the relocatable logical unit tables element generated during SYSGEN procedures.
4. DEL EQU 'n' - where 'n' (as in the LIBS manual) is the compressed code value, expressed in decimal or hexadecimal notation, of the control punches in column 1 of the "command" cards. For example, if ∇ is desired as the control character, DEL EQU X'55; must be specified, and the punch is 12-0-2.

It is possible to use the same card as in a LIBS run deck to cross-reference a module by equating DEL to the same value as when LIBS was linked.

It is not necessary to link any control stream element to the program, since it is part of the main cross-reference element - DXREF.

Cards are read from the control stream device as specified at systems generation time.

When linking the cross-referencer, DXREF must always appear last.

EXAMPLE: (a control stream linker deck run)

```
/      EXEC      DL11
/      DATA     C
      CTL        card
      PRGM       XREF,*
      LIBE       NONE
DEL    EQU       X'55' (or DEL EQU 85)
      INCLUDE   /TPRYTR
      INCLUDE   /PLUN
      INCLUDE   /DXREF
/*
```

The disc cross-reference routine will fit into 16K.

Symbols, when cross-referenced will appear in alphabetical order on the printout.

The cross-referencer uses all memory from the end of where the program is linked to the end of available memory for storage of symbols, their references, etc.

The contents of the boundary table will be adjusted to reflect the memory available to the cross-reference routine when symbionts are used.

NOTE: It is advisable to run the cross-referencer without symbionts because of memory limitations, and to prevent the cross-referencer from recycling itself and making more passes at the input source code than really required.

C. RESTRICTIONS:

At this time, it is impossible to cross-reference a group. Each element to be cross-referenced must be specified.

D. ERROR HANDLING

DIAGNOSTIC	MEANING	ACTION
INVALID COMMAND	The command card does not have a valid control character in Col. 1. The command specified is invalid.	The command is ignored. If not at end-of-file, a new card is read.
TOO MANY GROUP NAMES	Group levels have exceeded the allowable 5.	The command is ignored. If not at end-of-file, a new card is read.
MODULE NAME MISSING	No group or module name specified.	The command is ignored. If not at end-of-file, a new card is read.
INVALID L.U. NO.	The logical unit number is invalid.	The command is ignored. If not at end-of-file, a new card is read.
UNIT IS DOWN	The physical unit table entry indicates the unit is not available.	The command is ignored. If not at end-of-file, a new card is read.
MODULE NOT ON TAPE ^①	The module-name cannot be found on the tape specified, or within the specified group-name. Tape mark read - no more modules.	The command is ignored. If not at end-of-file, a new card is read.
END STATEMENT MISSING	Assembler program with a missing END record. (Will occur on all PROC's)	If a PROC ignore the error. If not a PROC, check for missing END record. In either case, processing continues.
TABLE OVERFLOWED	Not enough memory to cross-reference entire source coding in one pass.	Processing continues with recycling, until all source coding for that module is processed.
NODEF	Undefined symbol	Processing continues, but check symbols against references.

DIAGNOSTIC		
DUPLICATE	Duplicate symbols	Processing continues, but check duplicate symbols against each other.
UNREFERENCED	Unreferenced symbol	Processing continues, but check if symbol can be deleted.
****TAPE I/O ERROR**** ①	Tape IOCS Halts (20ux)	No diagnostic will be issued if a non-zero keyin is made to retry. Press START. If no keyin is made, the diagnostic will be given, and the source that has been processed up to this point will be cross-referenced. If not at end-of-file a new card will be read and processing continues. See 9200/9300 HALT/DISPLAY Manual.
30ul ②	Program not found on physical unit # specified.	Press START switch to cancel.
14uX ②	8411/14 LOCATOR/ LOADER HALTS	Press the START switch to cancel; or make a non-zero keyin to retry. See 9200/9300 HALT/DISPLAY Manual for further explanation of halts.

① TAPE ONLY

② DISC ONLY

E. SPECIAL NOTE: RECYCLING

When the symbol table has filled all available memory, recycling occurs if more symbols are to be processed. As a result, certain symbols may have "***NODEF**", or "***UNREFERENCED***" printed beside them. These symbols should be checked since their definitions and/or references may appear on subsequent or previous recycle passes.

F. HARDWARE CONFIGURATION:

The 9200II Configuration must be equipped with the hardware multiply/divide/edit features.

G. DOCUMENTATION:

Changes will be made to the Disc SS UTILITY Manual, UP-7835, Rev. 2 to reflect this new feature.

APPENDIX R. NOTE JOB CONTROL CARD

The purpose of the NOTE card is to insert comment cards in the System Job Control Stream (SJCS), or whenever a comment card is needed in a Job Control Stream deck.

The format of the NOTE control card is:

1 10 16
/ { } NOTE COMMENTS

Should Contain
Spaces (See Appendix S)

On reading a NOTE card, Job Control will ignore it and read the next card. However, if the SKIP command is in force, columns 3-8 of NOTE cards are checked for the label being SKIPPed to.

Comments are allowed in columns 16-80.



APPENDIX S. SKIP JOB CONTROL CARD

The SKIP command has been expanded. Positional parameter 1 has been defined and positional parameter 3 has been added. The format of the expanded SKIP command is:

LABEL	OPERATION	OPERAND
1	1Ø	16
/	SKIP	[LABEL] , [[X] [Ø] [1]] , [[N] [P] [X]] [[N] [P] [X]] [[N] [P] [X]]

The function of the SKIP command card is to make a test between the SKIP byte, as specified by positional parameter 2, and the UPSI byte (location X'Ø117') and if the test makes the condition(s) specified by positional parameter 3.

1. Cancel if LABEL (positional parameter 1) is unspecified or
2. Bypass images in the control stream until a control card image (/ in col 1) is detected which contains the matching label at which point control stream processing is resumed. However, if a matching label is not detected all input cards are exhausted by the card reader (X'6140') and all control cards including FINIS, DVCUP, DVCDN and JOB are ignored.

NOTE: If Cancel takes place and

1. The control stream was being read from the system medium, the next (and subsequent) images will be taken from the card reader. At this point control stream processing continues as in the next paragraph.
2. The control stream was being read from the control stream card reader; all control cards except JOB, FINIS, DVCUP and DVCDN will be ignored. JOB and FINIS will stop the cancel procedure and DVCUP and DVCDN will mark units up and down respectively.

The test between the SKIP byte (positional parameter 2) and the UPSI byte is made by a Test Mask command with SKIP byte as the I2 value and the location of the UPSI byte as the OP1 value. The results of the test can be controlled by the specification for positional parameter 3. The results of this test can be defined as:

1. Non-match - no one bits of the SKIP byte have matching one bits in the same position of the UPSI byte
2. Partial Match - some one bits of the SKIP byte have matching one bits in the same position of the UPSI byte.
3. Full Match - all one bits of the SKIP byte have matching one bits in the same position of the UPSI byte.

The specifications for the three positional parameters are:

Positional Parameter 1 LABEL

A label is from one to six characters in length and may be comprised of any of the 256 available combinations of punches, except comma (0-3-8) and space (no punch). The use of a label implies that a following control card will have a matching label punched in columns 3-8.

Labels in excess of 6 characters will be truncated to the most significant 6 characters.

■ Positional Parameter 2 $\left\{ \begin{matrix} X \\ \emptyset \\ 1 \end{matrix} \right\} \dots \left\{ \begin{matrix} X \\ \emptyset \\ 1 \end{matrix} \right\}$

Positional parameter 2 is the SKIP byte and is represented in the same manner as the UPSI byte of the UPSI control card. If less than 8 bits are specified the least significant bits are assumed to be zero(s).

Any representation other than 1 is treated as \emptyset .

■ Positional Parameter 3 $\left[\begin{matrix} N \\ P \\ X \end{matrix} \right] \left[\begin{matrix} N \\ P \\ X \end{matrix} \right] \left[\begin{matrix} N \\ P \\ X \end{matrix} \right]$

Positional parameter 3 specifies what test results of the SKIP byte/UPSI byte test are to cause a skip (to the label specified by positional parameter 1) or cancel (no label specified).

The specifications may be used individually or in combination in any sequence. They are:

- N - a non-match condition will cause skip/cancel
- P - a partial match condition will cause skip/cancel
- X $\left[\begin{matrix} \text{Not} \\ N \text{ or } P \end{matrix} \right]$ - a full match condition will cause skip/cancel

Positional parameter 3 will be examined for up to 8 characters. Examination is terminated by the first blank position in a scan starting at the most significant position.

The default (no positional parameter 3 specified) test is full match.

■ Comments

Positional parameter 3 must be separated from any comments by at least one blank column.

Examples 1-16 which follow show some variations of the SKIP command card form and its use and are commented on below. In these examples, comments X is used to represent either zero or 1.

COMMENTS

- Ex. 1. An UPSI value of 1XXXXXXXX will cause a skip to label ABCDEF.
- 2. An UPSI value of \emptyset XXXXXXXX will cause a skip to label ABCDEF.
- 3. An UPSI value of 1XXXXXXXX will cause a skip to label ABCDEF.
- 4. If the UPSI value is not X1X1XXXX a skip to label X will take place.
- 5. An UPSI value of X1X1XXXX or X \emptyset X \emptyset XXXX will cause a skip to label X.
- 6. An UPSI value of X111XXXX will cause a cancel.
- 7. An UPSI value of X \emptyset \emptyset XXXX will cause a cancel.
- 8. An unconditioned skip to label Y will take place.
- 9. An unconditional cancel will take place.
- 10. An unconditional cancel will take place.

11. An UPSI value of 11XX11XX will cause a skip to label A1B2C3 to take place.
NOTE: 1. truncation of label to six (6) positions
2. none of the 'conditions' specified in positional parameter 3 are N or P and therefore a full match only is assumed to be the test desired.

12. An UPSI value of 00XX00XX will cause a skip to label A1B2C3.
NOTE: see note 1 in example 11.

13. An UPSI value of X1XX1XXX or X0XX0XXX will cause a skip to label ?!,.:% to take place
NOTE: the 'N' of GONE and the 'non-N' and 'non-P' of GONE are the same as specifying NF or FN.

14. If the UPSI value is not 00000000 a skip to label 012345 will take place.

15. A test is made at line 7 to determine if the assembler set UPSI bit 5 because of assembly errors. If bit 5 is not set a skip to label A (line 10) will take place where the linking of the program will take place.

If bit 5 is set, a halt/display of '0001' will occur because of the PAUSE card (line 8). If UPSI bit 5 is manually reset (set to 0) at this time and run is depressed, the test at line 9 will fail and the program will be linked.

If UPSI bit 5 is not manually reset at this time the test at line 9 will be successful and a cancel will take place.

An identical test is made in lines 14-16 on the results of the linker except, of course, UPSI bit 4 is the controlling bit.

16. Example 16 is a variation on example 15. The test for any assembler or linker errors is made in line 11. If UPSI bits 4 and 5 are not set (UPSI is XXXX00XX) a skip to label A will take place and the program will be executed.

If UPSI bits 4 and/or 5 are set (UPSI is not XXXX00XX) halt/display '0001' will occur because of the PAUSE card at line 12. If UPSI bits 4 and 5 are manually reset at this time and run is depressed, the test at line 13 will fail and the program will be executed.

If UPSI bits 4 and 5 are not reset at this time, the test at line 13 will be successful and a skip to label B will take place. At that point one of two things will occur:

1. If the control stream was coming from the control stream reader a FINIS loop will be entered.
2. If the control stream was coming from the system medium, the next control stream image will be taken from the control stream reader.

NOTE: In case 2, if line 13 was potentially a cancel (/ SKIP ,000011,PF) instead of a skip to label B an immediate reversion to the control stream reader would take place and that control stream would be bypassed until a JOB or FINIS control card was detected. That is, the cancel would be applied against the control stream reader.

E
X
A
M
P
L
E

	LABEL	OPERATION	OPERAND
	1	10	16
1	/	SKIP	ABCDEF,1
2	/	SKIP	ABCDEF,1,N
3	/	SKIP	ABCDEF,1,PF (OR FP)
4	/	SKIP	X,Ø1Ø1,NP (OR PN)
5	/	SKIP	X,Ø1Ø1,NF (OR FN)
6	/	SKIP	,Ø111 (FULL MATCH ASSUMED)
7	/	SKIP	,Ø111,N
8	/	SKIP	Y,,N
9	/	SKIP	,,N
10	/	SKIP	,,FNP (OR ANY SEQUENCE OF N, P, F)
11	/	SKIP	A1B2C3D4,11ØØ11ØØ,JUMTOTAG
12	/	SKIP	A1B2C3D4,11ØØ11ØØ,N
13	/	SKIP	?!;.:%,X1ØØ1X,GONE
14	/	SKIP	Ø12345,11111111,PF (OR FP)

EXAMPLE 15

L I N E	LABEL	OPERATION	OPERAND
	1	10	16
1	/	JOB	SKIP TO A LABEL EXAMPLE - 14 FEB 75
2	/	DATE	'750214'
3	/	EXEC	KSMB
4		.	
5		.	
6	/*		
7	/	SKIP	A,000001,N IF NO ASSEMBLY ERRORS TO EXEC DL11
8	/	PAUSE	0001 IF ANY ASSY ERRORS, DISPLAY '0001'
9	/	SKIP	,000001 IF UPSI BIT 5 IS NOT RESET, CANCEL
10	/ A	EXEC	DL11 UPSI BIT 5 WAS RESET OR NO ASSY ERRORS
11		.	
12		.	
13	/*		
14	/	SKIP	B,00001,N IF NO LINKER ERRORS TO EXEC LINKED PROGRAM
15	/	PAUSE	0002 IF ANY LINKER ERRORS, DISPLAY '0002'
16	/	SKIP	,00001 IF UPSI BIT 4 IS NOT RESET, CANCEL
17	/ B	EXEC	PROGRAM UPSI BIT 4 WAS RESET OR NO LINKER ERRORS
18		.	
19		.	
20	/*		
21	/	FINIS	FINIS LOOP AT END OF LINKED PROGRAM -OR IF CANCEL

EXAMPLE 16

L I N E	LABEL	OPERATION	OPERAND
	1	10	16
1	/	JOB	SKIP TO A LABEL EXAMPLE - 14 FEB 75
2	/	DATE	'750214'
3	/	EXEC	KSMB
4		.	
5		.	
6	/*		
7	/	EXEC	DL11
8		.	
9		.	
10	/*		
11	/	SKIP	A,000011,N IF NO ASSY OR LINK ERRORS, TO EXEC PROGRAM
12	/	PAUSE	0001 IF ANY ASSY OR LINK ERRORS, DISPLAY '0001'
13	/	SKIP	B,000011,PF IF UPSI BITS 4 & 5 NOT RESET, TO FINIS
14	/ A	EXEC	PROGRAM UPSI BITS 4 & 5 RESET OR NO ASSY OR LINK ERRORS
15		.	
16		.	
17	/*		
18	/ B	FINIS	

APPENDIX T . VOL JOB CONTROL CARD

The purpose of the VOL card is to check and verify the volume serial number of the disc on the logical unit number specified.

The format of the VOL control card is:

```
1      10      16
/      VOL    LU#,Disc Serial#bCOMMENTS
```

■ Positional Parameter 1

LU# - is the logical unit number of the disc unit expressed as a two-digit hexadecimal number.

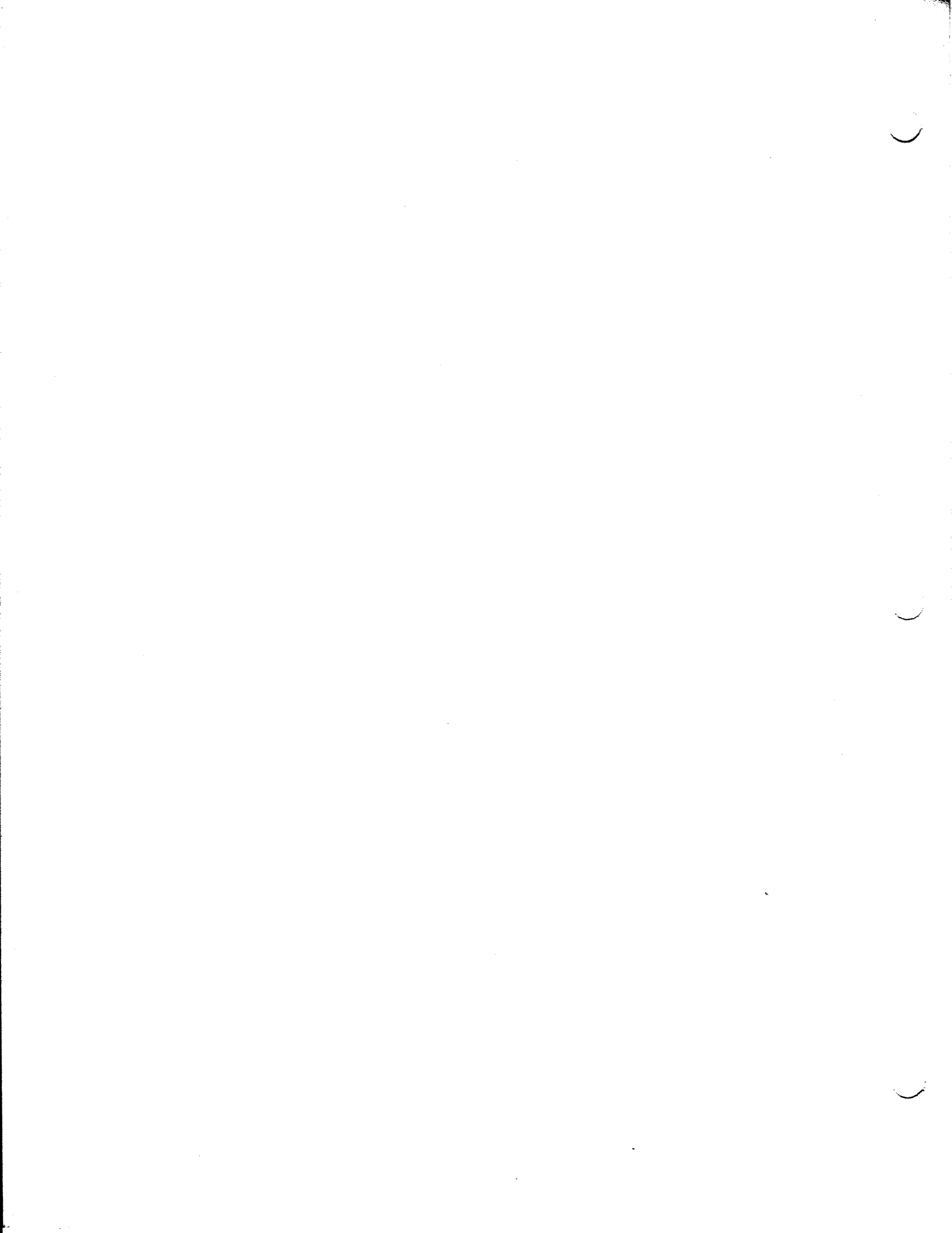
■ Positional Parameter 2

,Disc Serial# - is the alphanumeric EBCDIC representation (6-character maximum length) of the unique code assigned to the disc pack when DASMed. This number is resident on the disc pack in the Standard Volume Label (VOL1).

The halt/displays issued will be based upon the following conditions:

<u>HALT/DISPLAY</u>	<u>CONDITIONS</u>	<u>ACTION</u>
412F	<ol style="list-style-type: none">1. logical unit number is not assigned to a disc;2. logical unit number is expressed as a one-digit number;3. logical unit number is outside the LU/PU table range	See the HALT DISPLAYS MANUAL, UP-7719 Rev. 4.
41Du	The disc volume serial numbers do not match on <u>physical</u> unit 'u'.	To retry: mount a different disc pack on the physical unit u specified. Press START. To cancel: make an external cancellation keyin. (Operator request of X'F0'), Press START.
41Eu	Any disc device status other than a successful find on <u>physical</u> unit 'u' for the VOL1 label.	For recovery procedures, see 41Du.

Note: The VOL card should be inserted after JOB and SWAP control cards to obtain valid results.



APPENDIX U. SEQUENTIAL DISC IOCS (DSPIT and DTFSD).

1. Extensive changes were made to both DSPIT and DTFSD to provide faster assembly and more comprehensive error checking. Both must be assembled by the new (5.0 or higher level) assembler.
2. Corrected errors in DSPIT's handling of variable length records which often resulted in data being lost or scrambled or in records with incorrect lengths being generated. The following information may clear up some confusion that exists on the subject of variable length records.
 - a. When delivering a variable length record to the IOCS for output, the first two bytes must contain the size of the data portion of the record, plus four. The actual data content of the record begins at record plus four bytes. The second two bytes are not used by 9300 sequential IOCS but are zero-filled by the IOCS for compatibility with other systems.
 - b. Upon input, the program receives the record in the same form in which it would write it out, i.e. with record size and system bytes in the first four bytes of the record.
 - c. In generating DTFSD, RCSZ does not need to be specified for either an input or output file since the user program supplies and receives record size in the first two bytes of each record.
 - d. If variable length unblocked records are being written, the IOCS appends an additional four-byte field to the beginning of each record. The two bytes contain the physical block size (total record size plus four); the next two bytes are not used by 9300 IOCS but are cleared to zero. Therefore if an unblocked variable-length record contained 100 bytes of user data, it would occupy 108 bytes (plus 8-byte count) on disc.
 - e. If an RPG program wishes to write variable-length records, it must insure that the first two bytes of each record contain the record size, as described in "a" above. For instance if a record to be written contained 75 bytes of data, a statement such as the following would have to appear in a calculation specification card:

```
C      MOVE X'004F' RECSZE      2
```

and a statement such as the following in an output specification card:

```
0      RECSZE      2
```

The beginning of the actual data portion of the record would be considered to be byte 5 for purposes of filling out input and output specification cards.

- f. If programming in COBOL, each possible length of record to be written should have its own 01 level entry in the file descriptions. The length of the four extra bytes at the beginning of each record must not be added in; the COBOL compiler generates code which causes necessary conversion to take place.
- g. Errors formerly occurred if the work area from which a variable-length record was being written was not halfword-aligned. This is no longer the case. Work areas do not have to be halfword-aligned; all I/O areas (IOA1 and IOA2) must be halfword aligned for any record format.

- h. The use of a work area for writing variable-length blocked records is not mandatory but is highly recommended. If you choose to deliver variable-length blocked records in the I/O area via the IORG register, an additional area following the I/O area must be reserved which is as long as the largest record which might be written. When a record is delivered to the IOCS which would cause the blocksize to overflow the maximum specified by the BKSZ parameter, the IOCS writes out whatever data is in the I/O area, then moves the current record to the beginning of the I/O area. No VBLD parameter or TRUNC macro exists, as in Tape IOCS.
3. When an IORG register was specified with fixed-length unblocked records and a single I/O area, sequential IOCS formerly failed to load the register with the address of the current input record. The use of IORG in such a situation is optional; if the programmer specified IORG in DTFSD, the register specified will be loaded with the correct address.
4. The capability now exists of adding to a sequential disc file. For purposes of this discussion, "adding" consists of the ability to start writing new records at the address currently occupied by the end of file mark instead of being forced to start any output file at the beginning of the file. The following are some of the programming considerations involved in the use of this feature:
- a. To define a file as being added to, the following must be done:
- (1) In Assembler, include ADD=YES in the parameters specified when generating DTFSD.
 - (2) In RPG, punch an "A" in column 66 of the "F" card describing the sequential file which is to be added to.
 - (3) In COBOL, no direct means exists of specifying addition, but the file table of the file in question may be modified either by a PARAM card or by an Assembler subroutine. In the compilation listing's data division map, the address of each disc file table is given. The contents of byte ten of the file table (table plus X'0A') must be changed from 00 to 04. Byte eleven always contains zeroes initially. Therefore, to modify a COBOL program whose sequential file table began at X'2258' by use of a PARAM card:


```

/          EXEC  COBPROG,01,LOADLIB
/          PARAM R2262:0400
      
```

The modification of the program could be done selectively with the PARAM card omitted when it was desired to start at beginning of file. Assembler programmers may also wish to use this technique but should remember that the file table begins at "filename" plus 12 bytes, with "filename" being the tag placed on the DTFSD macro instruction.
- b. All processing which is directly related to the add function is performed in the open and close transients. Any existing program may be modified to add to an output file simply by changing the contents of byte ten of its file table. Any existing program which creates an output file, if run with the new (5.0 and higher) transients, will now insert the necessary pointers in the file's Format-1 label when it CLOSEs the file. This applies to programs written in any language.

c. The sequential IOCS now uses bytes 98 through 102 of a sequential file's Format-1 label to preserve the pointers which will be used when the file is to be added to. The CLOSE transient updates the Format-1 to include the pointers, whether or not the file was being added to. The OPEN transient, if the file is an add file, uses the pointers to locate the track and record where output is to begin. It is therefore just as important that a SAM file be CLOSED as an ISAM file. If a program aborts before it can CLOSE a SAM file, the Format-1 pointers are as they were when the program began execution; if other functions of the program which aborted can be safely rerun, the sequential file can be recovered by rerunning the program from the beginning.

d. The format of the pointers in the Format-1 label is "ttrll" with:

- tt being the relative track number (starting with X'0001' as the first track of the file) of the track containing the end-of-file mark.
- r being the record number of the end-of-file mark
- ll being the number of bytes remaining on this track. This value is necessary for the IOCS to resume writing data where it left off.

Since "tt" is a relative value, the file may be copied to different physical extents by the Dump/Restore Utility (DPRS) without losing its capability of being added to. DPRS preserves this information when dumping to tape, restoring from tape, and dumping disc-to-disc.

e. If an output file has been defined as an add file but bytes 98 through 102 of its Format-1 label contain binary zeroes, a 23u0 display is given. The following replies may be made to this display:

- 00 - To retry reading the Format-1 label. Could be used if the wrong pack were mounted.
- 01 - To cancel the job.
- any other reply - To start writing output data at the beginning of the file.

f. When each volume of a multiple-volume file is closed, the IOCS clears bytes 98 to 102 of the Format-1 label to zeroes to indicate that it is not possible to add to this volume of the file. The last volume to which data was written must be mounted for adding to proceed.

g. If it is necessary to modify an RPG program to add to a file, the address of the file table may be ascertained as follows:

- (1) Mount a disc pack which does not contain the desired file on the correct logical unit.
- (2) Execute the program which would write the file.
- (3) When 23u4 display occurs, the address of the file table is in PROC register 13 (locations 002A-2B). To this address must be added ten bytes, as described in 4.a.(3) above, to obtain the address of the byte that must be changed from 00 to 04. To verify this, a dump may be taken at the 23u4 halt; RPG generates file tables at the top of memory with each preceded by a series of 47F0 instructions similar to those generated by DTFSD in assembler.
- (4) If the same logical unit contains other files defined in the same program, this procedure may become complicated or impossible without some trial-and-error attempts.

5. If a zero reply was given to a 23uD halt, the IOCS did not retry. This minor discrepancy has been corrected. IOCS will only ignore a creation/expiration date error if a reply of greater than one is given.
6. The new parameter NONT is available in the DSPIT macro. If NONT=YES is specified, the sequential IOCS which is generated will have no transient capabilities whatsoever. Such a generation would only be desirable if the IOCS so generated were to be used in an MOS environment or within a symbiont. Although the obsolete "non-transient" SAM macros (DSP11 and DSP12) are still available with this release, they will not be distributed with any future DNCOS releases. No changes have been made to DSP11 or DSP12 in four years.
7. The sizes of most possible generations of sequential IOCS routines have been reduced, in some cases by several hundred bytes.
8. The sequential IOCS routines which process input and output table files in RPG programs have been brought up to the current level of the SAM processor.

APPENDIX V. 8411/8414 DIRECT PROCESSOR

The Direct Processor, comprised of the macros

DTFDH - Define the File Direct Handler and
DAMIT - Direct Access Method I/O Transient

is a processor designed to allow the user a control over his files that is closer to physical IOCS than either the Sequential or Index Sequential Access methods.

The Direct Processor is not a 'random' processor, per se, but allows the user to establish any form of file access/maintenance he desires.

Certain constraints are imposed, e.g. VOL1, Format 4, Format 1, and Format 3 label checking; only in the interest of protecting the user of the direct processor from himself, particularly during debugging time.

The Direct Processor allows for three different methods of accessing a file:

1. ID - by address, either absolute or relative
2. KEY - find by key from a starting address
3. AFTER - write a new record after the last record at a given address

It also allows for three different ways of addressing a file:

1. Absolute - fixed address including volume no.
2. Relative Record - range 1 - 65535
3. Relative Track plus absolute record - range 1 to limit of discs.

Although a given file can be accessed in any combination of ID, KEY and AFTER, it can only be addressed in one of three ways in one DTFDH module.

Only one DAMIT module is allowable and necessary in one program and through keyword parameters may be tailored to cover the necessary options in a minimal amount of memory.

One DTFDH module is necessary for each file to be processed by the Direct Processor although any file may be accessed by more than one DTFDH module. Other than memory constraints, there is no limit to the number of DTFDH modules that may be included in any one program.

The Direct Processor is not directly available to card assembler users. It must be assembled using the Tape/Disc assembler and the relocatable result of the assembly punched out and used as input to the card linker.

The DAMIT may be assembled

1. Non-transient or
2. Transient

When assembled with transient capability OPEN and CLOSE are arbitrarily transient and KEY, ID, AFTR and IORG are optionally transient.

The DTFDH module is non-transient only and needs no notification that it is working with a transient or non-transient DAMIT.

The sizes of the two modules are as follows:

DTFDH $234+(10*(\text{number of volumes in file}))=244$ (minimum)

DAMIT	non-trans	Open/close Trans only	All trans
Max Input/Output	< 3800	< 2700	< 1500
Max Input only	< 2800	< 1700	< 1300

The FOR control card in DASM can be used to pre-format the various volumes of a direct access file in a fashion similar to the control cards of an ISF run for an ISAM file. If pre-formatting is desirable, the DR parameter of the FOR card can be used even when relative record (REL=R) is not specified in DAMIT and/or DTFDH. The above is true for any file to be processed by the direct processor when the record size is fixed, (RCFM=FIXUNB).

If record size is not fixed (RCFM=UNDEF) or for any reason it is not desirable to pre-format the disc, then the only way a file can be loaded is by using the WRITE AFTER(,EOF) and WRITE RZERO imperative macros.

NOTE: The WRITE AFTER(,EOF) and RZERO imperative macro not only update or reset the capacity record (RØ) but clear the balance of the track after the record just written (all of the track after RØ when WRITE RZERO is used).

While files may be keyed or unkeyed the key size of any given file must remain constant throughout the file.

In unkeyed files KLEN=Ø. In keyed files key size may range from 3 to 255.

EOF records (data length=Ø) have a key in keyed files KLEN bytes long.

Keyed files can be updated without reference to the key and also the key (not its size) can be changed.

For 8411, record size (RCSZ) may range

$$2 < \text{RCSZ} \leq 3605 - (\text{KLEN} + 20 \text{ if } \text{KLEN} \neq \emptyset)$$

For 8414, $2 < \text{RCSZ} \leq 7249 - (\text{KLEN} + 45 \text{ if } \text{KLEN} \neq \emptyset)$

Some restrictions that existed under DTFRA and DAMIO are no longer pertinent. They are:

- a. File accessing - files may now be accessed by ID and/or KEY and/or AFTR in any combination.
- b. Creation Date and Expiration Date are required only on output (TYPF=OUTPUT) files.
- c. Creation Date/Expiration Date check is optional on input files (TYPF≠OUTPUT).
- d. I/O areas (IOAL) need not be specified individually for each file. They may be shared between files. The validity of their content if shared is under user control.
- e. Not specifying AFTR=NO when RCFM=UNDEF is no longer an error. These two keyword parameters are independent of each other.
- f. Please note the changes in IORG content.
- g. Please note that there is no User Label handling facility in DTFDH/DAMIT at present.
- h. A further explanation of the two RCFM keyword specifications:

FIXUNB - the data is fixed in size on the disc. If blocked it must be unblocked and reblocked by the user as necessary.

UNDEF - the data is unknown in size or variable in size or both.

Obviously FIXUNB data may be processed as UNDEF if it is desirable for any reason. The converse is not true.
- i. TYPF parameter specifications have been changed to the following:

≠OUTPUT - file type is input and no writing of any kind will be allowed in this file.

=OUTPUT - file type is output and may be input also. All read/write orders allowed by other parameters will be allowed.
- j. Write verification is optional and the default is:

YES, if TYPF=OUTPUT or
NO, if TYPF≠OUTPUT
- k. A new imperative macro READ GKEY has been initiated. An explanation of its function will be found under READ in the imperative macro section.
- l. On a WRITE AFTER, EOF imperative macro, the capacity record (RØ) will no longer be updated to reflect the EOF record occupying a record position. A check will be made to insure the EOF record will fit (if it won't the return is to ERRO under the same conditions as any other record that won't fit). Therefore the next WRITE AFTER to this track will erase the EOF (extend the file/track) and correctly update the capacity record.

DIRECT ACCESS METHOD I/O HANDLER

DAMIT is an updated transientized version of DAMI1 and DAMI2 for use with DTFDH. It can be generated as a non-transient if desirable. See the keyword parameter NONT.

DAMIT will theoretically handle an infinite number of files (DTFDH modules). It must be generated to include all options specified in each of the DTFDH modules it will support.

The keyword parameters are: (all are optional)

ABS - default is YES

Valid Specifications - =NO, =YES, =∅

NO - SKAD symbol will contain relative addressing only. No absolute format.

YES } - SKAD symbol will contain absolute addressing for at least
∅ } one file.

(see NOTE under REL)

AFTR - default is YES

Valid Specifications - =NO, =YES, =∅, =TRAN

NO - No format write commands will be issued (WRITE AFTER(,EOF) and WRITE RZERO).

TRAN - Format write commands will be issued and they will be processed by a transient.

YES } - Format write commands will be issued and they will be
∅ } processed by a resident routine.

NOTE: If AFTR≠NO, TYPF=OUTPUT must be specified.
REL=R is invalid, unless ABS≠NO.

ID - default is YES

Valid Specifications - =NO, =YES, =TRAN, =∅

NO - No READ or WRITE ID macros will be issued.

TRAN - READ and/or WRITE ID macros will be issued and they will be processed by a transient.

YES } - Same as TRAN, except the processing routine is resident.
∅ }

IORG - default is YES

Valid Specifications - =NO, =YES, =TRAN, =∅

NO - User does not have a memory location reserved for the absolute address of finds by ID or KEY in any of the files being handled by DAMIT.

TRAN - User wishes to have absolute address of finds by ID and/or KEY presented to him and the routine to do this processing is a transient.

YES } - Same as TRAN, except the processing routine is resident.
∅ }

KEY - default is YES

Valid Specifications - =NO, =YES, =TRAN, =∅

NO - No WRITE or READ KEY (or GKEY) macros will be issued.

TRAN - WRITE and/or READ KEY (or GKEY) macros will be issued and the routine to process them is to be transient. REL=R is invalid unless ABS≠NO.

YES } - Same as TRAN, except the processing routine is to be resident.
∅ } REL=R is invalid unless ABS≠NO.

LBAD - default is NO

Valid Specifications - =NO, =TRAN, =∅

NO } - User header labels will not be processed.
∅ }

TRAN - User header labels will be processed and the routine that processes them will be transient.

NONT - default is NO

Valid Specifications - =YES, =NO, =∅

NO } - A non-transient DAMIT is not to be generated.
∅ }

YES - A non-transient DAMIT is to be generated. Any other keyword being specified =TRAN, where this specification would normally be valid, will be an error and prevent the generation of DAMIT. Other than =TRAN, all valid combinations of keyword parameter specifications are still valid.

Situations where "NONT=YES" would need to be specified would be

1. DAMIT is being generated for use within a symbiont, or
2. In a program which will operate in a Minimum Operating System (MOS) environment.

RCFM - default is FIXUNB

Valid Specifications - =UNDEF, =FIXUNB, =∅

UNDEF - Both UNDEF and FIXUNB records (or UNDEF only) will be processed.

REL=R is incorrect unless ABS≠NO.

FIXUNB } Only FIXUNB records will be processed.
∅ }

REL - default is YES

Valid Specifications - =NO, =R, =T, =YES, =∅

NO - SKAD symbol will only contain absolute addresses.

R - SKAD symbol will contain relative record addresses for at least one file.
AFTR≠NO and/or KEY≠NO and/or RCFM=UNDEF is incorrect unless ABS≠NO.

T - SKAD symbol will contain relative track addresses for at least one file.

YES } - SKAD symbol may contain relative record and relative track
∅ } addresses. (at least two different files)

- NOTE:
1. One file may be addressed in only one way; that is, it may be relative record or relative track or absolute.
 2. The ABS and REL keyword parameters determine how DAMIT will decode the SKAD symbol in the various DTFDH modules for which it is being generated.

SRCM - default is YES

Valid Specifications - =NO, =YES, =∅

NO - No file will be expecting to search beyond track limits on the READ or WRITE KEY (or GKEY) macros that are issued.

YES } - Some or all files will be searching on cylinder boundaries.
∅ }

NOTE: If SRCM≠NO, then KEY≠NO is incorrect, but not fatal to the generation of DAMIT; SRCM will be ignored.

TYPE - default is INPUT

Valid Specifications - =OUTPUT, =INPUT, =∅

OUTPUT - At least one file will have WRITE macros issued for it. All READ macros are allowed.

INPUT } - No WRITE macros will be allowed. All READ macros are
∅ } allowed.

If TYPF=OUTPUT, the creation date (CRDT) specified in an output (TYPF=OUTPUT) DTFDH module will be checked against the file expiration date and if the check passes, the expiration date (XPDT) specified will be written into the file Format 1 label as the new expiration date.

If TYPF≠OUTPUT, or an individual DTFDH module has TYPF≠OUTPUT specified then the creation date/expiration date check is optional.

- e.g. 1. If CRDT=symbol and TYPF≠OUTPUT in a DTFDH module the check will take place.
2. If CRDT=symbol is not specified and TYPF≠OUTPUT is specified then the creation date check will not occur.

NOTE: A 'good' creation date check is one in which the contents of CRDT symbol is greater than or equal to the contents of the expiration date in the Format 1 label of the file.

In Summary:

1. An output file must be supported by an output DAMIT and and specify a creation date and an expiration date.
2. An input file may be supported by any DAMIT and may specify a creation date. Expiration date (XPDT) will be ignored.

VERFY - default is =NO, if TYPF≠OUTPUT or
=YES, if TYPF=OUTPUT

Valid Specifications - =NO, =YES, =∅

NO - No WRITE macros will be check read for parity.

YES - All WRITE macros will include a check read for parity.

NOTE: If R∅ is written, it is always check-read. For example, in all WRITE AFTER(,EOF) and WRITE RZERO commands, Record ∅ will be check-read.

If a new Format 1 Label is written, it is always check-read.
(during CRDT/Expiration Date check)

WAIT - default is YES

Valid Specifications - =NO, =YES, =∅

NO - No WAITF imperative macros will be issued.

YES } - WAITF macros will be issued.
∅ }

All I/O imperative macros (not OPEN or CLOSE) for individual DTFDH modules where WAIT=YES is specified become merely head positioning (Seek) instructions. No data transfers will take place until the WAITF macro is performed. I/O imperative macros issued against individual DTFDH modules where WAIT≠YES is specified are unaffected by either specification for the WAIT keyword parameter in DAMIT.

DTFDH KEYWORD PARAMETERS

AFTR - enable use of format write commands

Optional - default is NO

Valid Specifications - =YES, =NO, =Ø

=YES is specified when it is the intention of the user to use any of the imperative macros; WRITE AFTER, WRITE AFTER, EOF or WRITE RZERO.

=NO } is specified when none of the above imperative macros will
=Ø } be used.

BKSZ - size of IOAL

Required

Valid Specifications - A decimal value in the range: greater than 2 and less than or equal to track capacity (+8, if AFTR=YES).

BKSZ specifies the size of the I/O area (IOAL).

The following factors determine the value of BKSZ:

1. The length of the data section of the disc record to be read or written. If variable in size (RCFM=UNDEF), it should accommodate the largest possible.
2. The length of the key section (KLEN=n) of the disc record to be read or written. This need only be considered if:
KLEN≠Ø and
AFTR and/or RDID and/or WRID is = YES
3. The length of the count section (8 bytes) of the disc record to be written if:

AFTR=YES

See IOAL Contents and IOAL Organization charts for the various make-ups of IOAL, and therefore the size (BKSZ) of IOAL. Minimum BKSZ specifications are:

KLEN + 11 if AFTR=YES
KLEN + 3 if AFTR≠YES

CRDT - creation date

Conditionally Required

Valid Specification - CRDT=symbol

CRDT=symbol is required if TYPF=OUTPUT. Symbol is the label of a 6-byte creation date to be checked against the expiration date of the file.

If TYPF≠OUTPUT the creation date/expiration date check is optional. See TYPF keyword parameter in DAMIT for a discussion on date check.

One possibility of the contents of CRDT symbol are YYYYMMDD where:

YY - year
MM - month
DD - day

in EBCDIC.

DEVx - Relative volume number and logical unit number of device for this file.

Optional - default is DEV1=Ø

Valid Specifications - DEVx=n where:

x - is the relative volume number in the range 1-8, and
n - is the logical unit number

DEV1 specifies the first volume of file, DEV2 the second, DEV3 the third, etc.

DVNO - the number of DEVx specifications that are valid.

Required

Valid Specifications - DVNO=n where n is in the range 1-8.

ERRO - user error routine

Required

Valid Specification - ERRO=symbol

Symbol is the label of the user error routine to which control will be returned under certain error conditions after the file has been opened error-free.

Error bits will be set in the Error Bytes of the DTFDH filetable. They are located at filename + 64₁₀ or filetable +16₁₀ or r15+16₁₀. Control may be returned to ERRO after any READ or WRITE macro.

See the Error Byte table for the contents and meaning of the various settings that may occur.

Error symbol must be on a half-word boundary.

FLID - file identification

Required

Valid Specification - FLID=symbol

Symbol is the label of a 44 byte file identification or label. Its contents must be identical to that specified in the LBL card when the file was DASM'd.

IOAL - I/O area

Required

Valid Specification - IOAL=symbol

Symbol is the label of a user area in which the disc I/O operations will take place. It is BKSZ bytes in length.

See the IOAL Contents and IOAL Organization charts for the make-up of the IOAL area.

IOAL must be on a half-word boundary if AFTR=YES.

IORG - I/O address register

Optional - default is no IORG.

Valid Specification - IORG=symbol

Symbol is the label of a 6 byte area located by the user which will contain the absolute disc address and volume number of the last disc access prior to termination of the CCW chain. This address is supplied by the direct processor at the conclusion of the last imperative macro. (not OPEN or CLOSE)

The address will be in the form, in binary:

MØCØHR where

M - relative volume number
ØC - cylinder
ØH - head
R - record

See the IORG chart for the representation of this address.

IORG must be on a half-word boundary.

KARG - key argument location

Conditionally Required

Valid Specification - KARG=symbol

Symbol is the label of a user area reserved to contain the key to be searched for during a READ KEY (or GKEY) or WRITE KEY imperative macro, and is KLEN bytes in length.

KARG=symbol is required only if RDKY or WRKY is = YES.

KLEN - key length

Conditionally Required

Valid Specification - KLEN=n where n is decimal and the range of n is $2 < n < 256$ and if the file was preformatted with the FOR card of DASM, n agrees with the key length specified in the FOR card.

KLEN=n is required only if RDKY or WRKY is = YES or

if the file has keys and AFTR=YES is specified, or

if the file has keys and REL=R

All keys in the same file must be the same length.

If only ID (READ ID, WRITE ID) processing is to be done it is not necessary to specify KLEN=n in files that are keyed unless:

It is desirable to update or examine the keys, or

Address the file in REL=R format.

LBAD - not available initially

RCFM - record format

Optional - default is FIXUNB

Valid Specifications - =UNDEF, =FIXUNB, =∅

=UNDEF is specified when the length of the data section is unknown and/or variable. This specification requires the specification RCSZ=r and precludes the use of relative record addressing (REL=R).

=FIXUNB } is specified when the length of the data section of the disc
=∅ } record is known and fixed in size. In this case the size
of the data section is BKSZ - (KLEN + (8, if AFTR=YES)).

RCSZ - record size

Conditionally Required

Valid Specification - RCSZ=r where r is a general register in the range $7 < r < 14$.

Reading - The RCSZ register will contain the length of the data section at the successful conclusion of the READ order. That is, it is supplied by the Direct Processor.

Writing - The RCSZ register will be loaded by the user with the length of the data section prior to giving the WRITE order (even if WAITF is used and WAIT=YES is specified).

RCSZ=r - will be ignored if RCFM≠UNDEF and must be specified if RCFM=UNDEF.

RDID - intention to use READ ID imperative macro

Optional - default is RDID=NO

Valid Specifications - =YES, =NO, =Ø

=YES - if KLEN≠Ø both key and data sections will be read when a READ ID imperative macro is issued. If KLEN=Ø only the data section will be read whether the file is keyed or not.

=NO }
=Ø } - No READ ID imperative macros will be issued.

RDKY - intention to use READ KEY or READ GKEY imperative macro

Optional - default is NO.

Valid Specifications - =YES, =NO, =Ø

=YES - Requires KLEN≠Ø, KARG=symbol is specified. Precludes the use of relative record addressing (REL=R) and implies that READ KEY and/or READ GKEY imperative macros will be issued.

=NO }
=Ø } - No READ KEY or READ GKEY imperative macros will be issued.

REL

Optional - default is no relative addressing (=Ø)

Valid Specifications - =R, =T, =Ø

=R - Cannot be used when RDKY, WRKY, or AFTR=YES. Cannot be used when RCFM=UNDEF. Specifies that SKAD symbol will contain a relative record address where the 1st record in the file is 1. If files are keyed, see the ID processing paragraph under KLEN.

=T - May be used under all conditions. Specifies that SKAD symbol will contain a relative track and:

for READ or WRITE ID - an absolute record number
for READ or WRITE KEY - a record number of binary zero
or all WRITE AFTER's
The 1st track of the file is track 1.

=Ø - May be used under all conditions. Specifies that SKAD symbol will contain a relative volume number and absolute disc address with the same consideration for record number as shown under REL=T.

SKAD - File Addressing Method

Required

Valid Specification - SKAD=symbol

Symbol is the label of an eight-byte area reserved by the user to contain the (starting) disc address at which the next imperative macro is to take place.

SKAD symbol must be on a half-word boundary.

When REL=R its format is $\emptyset\emptyset\emptyset\emptyset\emptyset rr$ where:

rr is the binary relative record number at which the next READ or WRITE ID will take place.

When REL=T its format is $\emptyset\emptyset\emptyset\emptyset\emptyset ttr$ where:

tt is the binary relative track number at which the next I/O operation will take place

r is the absolute record number for ID operations (READ and WRITE)

r is \emptyset for KEY (READ and WRITE) and all WRITE AFTER operations.

Note: in both relative record and relative track, the first record or first track in the file is 1.

When REL= \emptyset (or is unspecified) its format is $M\emptyset\emptyset\emptyset C\emptyset HR$ where:

M - is the binary relative volume number

C - is the binary cylinder number

H - is the binary head (track) number

R - is the binary record number

As in REL=T, R is the absolute record number for ID operations (READ and WRITE) or

\emptyset for KEY (READ and WRITE) and all WRITE AFTER operations.

SRCM - search multiple tracks

Optional - default is NO

Valid Specifications - =YES, =NO, = \emptyset

If SRCM=YES is specified and the volume on which the search for key is to take place has the file allocated on cylinder boundaries then the search for key (READ KEY, READ GKEY, WRITE KEY) will start at the beginning of the track in SKAD symbol and continue until a find is made or end-of-cylinder is detected by the hardware, whichever occurs first.

If SRCM=NO (or \emptyset) is specified or the volume on which the search is to take place does not have the file allocated on cylinder boundaries, then the search for key will be restricted to the track specified by SKAD symbol.

TABL- Extent Table

Required

Valid Specification - TABL=symbol

Symbol is the label of 16*n bytes reserved by the user to contain the extents and other pertinent file information. Sixteen bytes must be reserved for each extent of the file. n is the number of extents in the file. There is a maximum of 16 extents per volume and 8 volumes/file.

TABL symbol must be on a half-word boundary.

See DTFDH figure for the contents of the 16 byte Extent Table.

TYPF - Type of File

Optional - default is INPUT

Valid Specification - =OUTPUT, =INPUT, =Ø

=OUTPUT - The contents of the CRDT symbol will be checked against the file expiration date. The XPDT symbol contents will become the new expiration date of the file and all I/O macros may be issued against the file.

See TYPF keyword parameter in DAMIT for a discussion of date check when TYPF=OUTPUT.

=INPUT - No writing of any kind will take place or be allowed on this file. Only READ ID, READ KEY and READ GKEY imperative macros will be allowed.

VERFY - Verify Write

Optional - default is

=NO - if TYPF=OUTPUT
=YES - if TYPF=INPUT

Valid Specifications - =YES, =NO, =Ø

=YES - Write parity check will occur on all WRITE imperative macros.

=NO - Only file control records such as RØ, FORMAT1 will be verified when they are written.

WAIT - Wait for I/O Operation

Optional - default is NO

Valid Specifications - =YES, =NO, =Ø

=YES - On all READ and WRITE imperative macros only head movement (if necessary) will take place. Data transfers will not take place until the WAITF imperative macro is issued. That is, as soon as head movement has been initiated, control will be returned to the user.

~~=NO~~ } - Head movement and data transfer will be initiated upon issue of the READ or WRITE command and control will not be returned to the user until the issued CCW chain reaches termination.

No effect on OPEN/CLOSE macros.

See WAIT keyword parameter in DAMIT for a further discussion of the WAITF imperative macro.

WRID - Intention to use WRITE ID imperative macro

Optional - default is NO

Valid Specifications - =YES, =NO, =Ø

=YES - Specifies an intention to use the imperative macro
WRITE ID

=NO } - No WRITE ID imperative macros will be issued by the
=Ø } user.

WRKY - Intention to use WRITE KEY imperative macro

Optional - default is NO

Valid Specifications - =YES, =NO, =Ø

=YES - Specifies an intention to use the imperative macro
WRITE KEY.

=NO } - No WRITE KEY imperative macros will be issued by
=Ø } the user.

XPDT - Expiration Date

Conditionally Required

Valid Specifications - XPDT=symbol

If TYPF=OUTPUT, the XPDT symbol is required and will contain
the new expiration date for the file in which one possible
form is YYMMDD where:

YY - Year
MM - Month
DD - Day

in EBCDIC. (This is a standard form)

If TYPF≠OUTPUT, the XPDT symbol will be ignored and is not
required.

XTEN - File Extent Count

Optional

Valid Specification - XTEN=symbol

If it is desired by the user to obtain a count of the number
of extents in the file, he will specify XTEN=symbol and
DTFDH will present the number of extents in XTEN symbol at
the end of opening the file.

The count will be in binary and XTEN symbol must be on a
half-word boundary.

IMPERATIVE MACROS - DTFDH

I OPEN

The OPEN macro will open the file to allow accessing by the READ/WRITE macros. Validation checks will be performed on the VOL1, FORMAT4, FORMAT1 and FORMAT3 labels as required. File accessing as specified for DAMIT will be checked against the processing specified by the individual DTFDH files as they are opened to insure that DAMIT covers the options required by the DTFDH modules.

Format:

Label	Operation	Operand
name	OPEN	filename

filename - the symbol of the appropriate DTFDH macro; the file to be opened.

II READ

The READ macro reads a record from disc to the IOAl area. If WAIT=YES was specified, the WAITF macro must be issued to actually cause the data transfer to take place, as the READ will only cause arm movement to occur.

Format:

Label	Operation	Operand
name	READ	filename, ID KEY GKEY

Positional Parameter 1

filename is the symbol of the appropriate DTFDH macro.

Positional Parameter 2

ID - RDID=YES must have been specified in the DTFDH macro for this file. A read into IOAl will occur from the address specified by SKAD. If KLEN≠∅ both key and data will be read; if KLEN=∅, only the data section will be read, whether the file has keys or not.

KEY and GKEY - RDKY=YES must have been specified in the DTFDH macro for this file. A search for the key in the location KARG symbol will be initiated at the track specified by SKAD (REL≠R). If SRCM=YES and the volume has the file allocated by cylinder an unsuccessful search will run to end-of-cylinder; if either of these is not true, an unsuccessful search will occur at end-of-track.

A successful search (find) for the KEY positional parameter is equality; for GKEY it is when the key from the disc record is equal to or greater than the contents of KARG symbol. The key from disc will not be read into IOAl.

III WRITE

There are two forms of the WRITE macro. One performs format writes (count, key, data); the other performs non-format writes (key, data). In either case the writing of a key is under the control of the KLEN parameter of DTFDH, i.e. if KLEN=Ø, the writes are count, data; and data only respectively. The format writes also clear the balance of the track after writing the record.

The format writes are AFTER; AFTER,EOF and RZERO. All others are non-format writes.

If VRFY=YES was specified all written records will be check read for parity.

A. Non-format WRITE

Used to update a file in existence or create a file on volumes that have been pre-formatted; see FOR card in DASM. WRITE KEY cannot be used to create a file.

Format:

Label	Δ Operation Δ	Operand
name	WRITE	filename, { ID KEY }

Positional Parameter 1

filename - the symbol of the appropriate DTFDH macro

Positional Parameter 2

ID - WRID=YES must have been specified in the DTFDH macro for this file. The record in IOAL will be written at the disc address specified by SKAD for a length of

1. BKSZ, if RCFM=FIXUNB or
2. RCSZ+KLEN, if RCFM=UNDEF

If KLEN≠Ø both key and data will be written; if KLEN=Ø only the data section will be written, whether the file has keys or not.

KEY - WRKY=YES must have been specified in the DTFDH macro for this file. The record in IOAL will be written when a match is found for the key in KARG symbol, which will be located on disc at, or after, the address in SKAD symbol.

if SRCM=YES and the volume has the file allocated on cylinder boundaries, the search will continue until a match is made or end-of-cylinder is detected, whichever occurs first. If either of the above is not true, the search for match will be confined to the track specified in SKAD symbol.

B. Format WRITE

Used to clear, create and/or extend a file.

Format:

Label	Operation	Operand
name	WRITE	filename, { AFTER AFTER, EOF RZERO

Positional Parameter 1

filename - the symbol of the appropriate DTFDH macro

Positional Parameter 2

AFTER - a new record, contained in IOAL, is to be written after the current last record on the track specified in SKAD symbol (REL≠R). The new record (count, key (if KLEN≠0) and data) is written and the remainder of the track is cleared. The Capacity Record* (R0) will be updated to reflect the new last record and remaining recording space on the track.

If an examination of R0 shows insufficient space to record this record

1. The record will not be written and
2. R0 will not be updated

and return will be to the user ERRO routine with the error bytes of DTFDH reflecting the reason for entry into ERRO.

*Record Zero (R0) of each track of a file processed by WRITE AFTER(,EOF) imperative macros must contain in its data section a 'capacity record'. This capacity record is initialized by a WRITE RZERO command and is examined by each WRITE AFTER(,EOF) command issued against the track. See the Record Zero (Track Descriptor) Record layout for the content and format of this record.

AFTER, EOF - An EOF (data length=0) record is to be written after the current last record on the track specified in SKAD symbol. Except for the fact that R0 will not be updated to reflect the existence of the EOF record, it is processed the same as data being recorded with an AFTER positional parameter.

RZERO - clear the track specified in SKAD symbol and reset R0 to reflect the entire track as available for recording. This form can be used to clear a track or a file.

If WAIT=YES was specified in DTFDH then the WAITF imperative macro must be issued to cause writing in any form to take place. The WRITE macro will only cause head positioning.

IV WAITF

If the DTFDH macro was generated with WAIT=YES specified, then the WAITF imperative macro must be issued after all READ or WRITE macros to cause the data transfer to take place.

If WAITF macros are issued to a file in which WAIT=YES was not specified, return will be immediate.

Format:

Label	Operation	Operand
(name)	WAITF	filename

Positional Parameter 1

filename - is the symbol of the appropriate DTFDH macro.

V CLOSE

The CLOSE imperative macro initiates termination procedures for the file specified and marks it as closed. It must be re-opened before any READ's or WRITE's are issued.

Format:

Label	Operation	Operand
(name)	CLOSE	filename

Positional Parameter 1

filename - the symbol of the appropriate DTFDH macro; the file to be closed.

VI NOTES

On return to the user after all imperative macros

1. Registers 8-13 will be restored to their original contents, i.e. what they contained before the macro was issued.
2. Register 14 will point to the next instruction on the DTFDH filename table after the macro last was executed. See filename +0 through +48 in the DTFDH layout.
3. Register 15 will point to the 1st byte (FLID) of the file table for that DTFDH.

DIRECT PROCESSOR HALT/DISPLAYS

DISPLAY	MEANING	ACTION
2400	The options selected to process the file (DTF) and the I/O handling capabilities (DAMIT) are not compatible.	Any response is CANCEL.
24u1	One of the following errors has been detected on the file on Physical Unit u: 1. ID process and SKAD record no. is \emptyset 2. Invalid Vol. No. in an absolute address. 3. SKAD address is highest allowed. 4. SKAD record no. is $>$ track allows. 5. Unsupported imperative macro. 6. One of the required half-word aligned parameters is <u>not</u> on a half-word boundary.	Any response is CANCEL.
24u2	Attempting to OPEN an opened file or CLOSE an unopened file.	Any response is CANCEL.
24u3	The calculated record size is greater than the BKSZ parameter specified.	$\emptyset/1$ CANCEL $\neq(\emptyset/1)$ IGNORE
24u4	A CCW chain has a program error in it.	Any response is CANCEL.
24u5	An Imperative Macro has been issued without an intervening WAITF on a file expecting WAITFs to be issued.	Any response is CANCEL.
24u6	A CCW chain has returned a 'hardware error' status on an unopened file.	Any response is CANCEL.
24u7	An imperative macro has been issued against an unopened file.	Any response is CANCEL.
24u8	Volume sequence check has failed in opening a file.	\emptyset RETRY $\neq\emptyset$ CANCEL
24u9	The record found as a Format 4 (VTOC) record does not have an ID of '4'.	\emptyset RETRY $\neq\emptyset$ CANCEL
24uA	A bad dispatcher status (length error, EOF or no find) has been returned on the search for VOL1 or FMT4.	Any response is CANCEL.

cont.

DISPLAY	MEANING	ACTION
24uB	An error status of 'EOF' or 'Wrong Length' has been returned on the search for the Format 1 label.	Ø RETRY ≠Ø CANCEL
24uC	Cannot find the Format-1 label (File ID). The file is not on this pack.	Ø RETRY ≠Ø CANCEL
24uD	The record found as the Format 1 does not have an ID of 1.	Ø RETRY ≠Ø CANCEL
24uE	The Creation Date/Expiration Date Check has failed.	Ø RETRY ≠Ø CANCEL
24uF	Either bad status (either length error, EOF or no find) or the ID is not '3' on the search for a Format 3.	Ø RETRY ≠Ø CANCEL

2 of 2

DAMIT KEYWORD PARAMETERS

KEYWORD	DEFAULT	VALID	COMMENTS
ABS	YES	NO	No file is using absolute addresses. ①
		YES	1 or more files are using absolute addresses.
AFTR	YES	NO	No format WRITES will be issued. ⑤④③②
		YES	Format WRITES will be issued against 1 or more files. ⑤④③
		TRAN	
ID	YES	NO	No macros with ID positional parameter will be issued. ②
		TRAN	Imperative macros with ID positional parameter will be issued against 1 or more files. ⑤
IORG	YES	NO	No files have specified an IORG symbol.
		TRAN	1 or more files have specified IORG symbol.
KEY	YES	NO	No macros with KEY positional parameter will be issued. ②
		TRAN	Imperative macros with KEY positional parameter will be issued for 1 or more files. ③⑤
LBAD	NO	NO	NOT AVAILABLE INITIALLY
		TRAN	
RCFM	FIXUNB	UNDEF	Both types (UNDEF and FIXUNB) supported.
		FIXUNB	Only RCFM=FIXUNB files will be supported.
REL	YES	NO	No file is using relative addresses. ①
		R T	REL=R and $\left\{ \begin{array}{l} \text{KEY} \neq \text{NO} \\ \text{AFTR} \neq \text{NO} \\ \text{RCFM} = \text{UNDEF} \end{array} \right\}$ and ABS=NO is an ERROR.
SRCM	YES	NO	No multi-track searches will be initiated.
		YES	1 or more files expect to search multi-track (KEY)

cont.

KEYWORD	DEFAULT	VALID	COMMENTS
TYPF ⑤	INPUT	OUTPUT	WRITE macros allowed. Expiration date checked and updated.
		INPUT	No WRITE allowed. Optional date check and expiration date update.
VERFY	*	NO YES	*If VRFY is unspecified and 1. TYPF=OUTPUT, default is VRFY=YES or 2. TYPF≠OUTPUT, default is VRFY=NO
WAIT	YES	NO	No WAITF imperative macro will be issued.
		YES	WAITF macro will be issued against 1 or more files.

- ① ABS=NO and REL=NO is an ERROR
- ② AFTR=NO and ID=NO and KEY=NO is an ERROR.
- ③ See REL keyword.
- ④ Format WRITES are WRITE AFTER(,EOF) and WRITE RZERO
- ⑤ TYPF=OUTPUT must be specified if any WRITE macros are issued.

2 of 2

DTFDH KEYWORD PARAMETERS

KEYWORD	DEFAULT	VALID	COMMENTS
AFTR	NO	NO	No FORMAT WRITES will be issued against this file.
		YES	FORMAT WRITES may be issued against this file.
BKSZ	ERROR	MIN. REQD.	Number of bytes in IOAL area. Minimum required is: KLEN+11 if AFTR=YES; or KLEN+3 if AFTR≠YES
CRDT	REQUIRED IF TYPF=OUTPUT OPTIONAL IF TYPF≠OUTPUT	SYMBOL	Address of Creation Date - used to test against file Expiration date.
DEV _x	DEV1=∅	-	No. of devices in file and their logical unit number. That is, DEV1=7 means the 1st volume is located on L/U 7.
DVNO	ERROR	1-8(10)	The number of DEV _x specs. The number of volumes in the file.
ERRO	ERROR	SYMBOL	The address of the user Error Routine. See the Error Byte chart.
FLID	ERROR	SYMBOL	The address of the 44 byte file ID.
IOAL	ERROR	SYMBOL	The address of the input/output area. It is BKSZ bytes in size.
IORG	No IORG	SYMBOL	The address of the 6 byte area to which disc addresses will be returned by DAMIT. See IORG chart.
KARG	See RDKY, WRKY	SYMBOL	The address of the key in memory. It is KLEN bytes in size. Required for KEY processing only.
KLEN	No KEY	3-255(10)	The size of KARG and/or the key section on disc. KLEN is required on keyed files when REL=R, even if <u>not</u> KEY processing.

KEYWORD	DEFAULT	VALID	COMMENTS
LBAD	NOT AVAILABLE INITIALLY		
RCFM	FIXUNB	UNDEF FIXUNB	Defines the disc records as being known and all the same size (FIXUNB) or not (UNDEF)
RCSZ	See RCFM=UNDEF	8-13(10)	If RCFM=UNDEF, then RCSZ=n, where n is a general register (8-13). Not required or used when RCFM=FIXUNB.
RDID	NO	NO	No READ ID imperative macros will be issued.
		YES	READ ID macros may be issued against this file.
RDKY	NO	NO	No READ KEY (or GKEY) imperative macros will be issued.
		YES	READ KEY (or GKEY) macros may be issued.
REL	No REL	R	SKAD address will be relative record. No KEY or AFTR processing allowed.
		T	SKAD address will be relative track.
		No REL	SKAD address is absolute with relative volume number.
SKAD	ERROR	SYMBOL	Address of the 8 byte area containing the disc address. The format of SKAD is specified by REL.
SRCM	NO	NO	On KEY macros, no multi-track searching.
		YES	On KEY macros, multi-track search if volume is cylinder allocated.
TABL	ERROR	SYMBOL	Address of area 16 times number of file extents in size.
TYPF	INPUT	OUTPUT	All macros allowed. File Expiration date checked & updated.
		INPUT	Only READ macros allowed. File Expiration date optionally checked - not updated.

2 of 3

KEYWORD	DEFAULT	VALID	COMMENTS
VERFY	*	NO YES	*If VRFY is unspecified and 1. TYPF=OUTPUT, then default is VERFY=YES or 2. TYPF≠OUTPUT, then default is VERFY=NO
WAIT	NO	NO	No WAITFs will be issued. Complete disc I/O on READ or WRITE.
		YES	WAITF will be issued. Disc I/O completed on WAITF.
WRID	NO	NO	No WRITE ID imperative macros will be issued.
		YES	WRITE ID macros may be issued. TYPF=OUTPUT
WRKY	NO	NO	No WRITE KEY imperative macros will be issued.
		YES	WRITE KEY macros may be issued. TYPF=OUTPUT
XPDT	REQUIRED IF TYPF=OUTPUT	SYMBOL	Address of 6 byte Expiration Date, required if TYPF=OUTPUT
XTEN	NO XTEN	SYMBOL	User wishes a count of file extents placed in this 2 byte location.

3 of 3

ERROR BYTES - STATUS CODE

LOCATED AT FILENAME+6410 or r15+1610 ON RETURN TO ERRO SYMBOL

BYTE	BIT	DESCRIPTION
∅	∅	Not Used.
∅	1	Length Error. 1. Fixed-length (RCFM=FIXUNB) - any length discrepancy between the size specified by the count section of the disc and BKSZ parameter for READ or WRITE macros. See Note 1. 2. Undefined (RCFM=UNDEF) A. WRITE macros - any length discrepancy between the size specified by the count section of disc and the RCSZ plus KLEN parameters. See Note 2. B. READ macros - only if BKSZ is less than the size specified by the count section of the disc record.
∅	2	Not Used.
∅	3	Not Used.
∅	4	No room on track. A WRITE AFTER(,EOF) has been attempted on a track whose capacity record (R∅) shows insufficient space to write the record. R∅ has not been altered and no portion of the record has been written.
∅	5	Not Used.
∅	6	Not Used.
∅	7	Not Used.
1	∅	Not Used.
1	1	Not Used.
1	2	Not Used.
1	3	Data Check - hardware failure The READ or WRITE macro has not successfully terminated. Either ignore the error and continue processing or close the run and notify Sperry Univac field engineers.
1	4	Record Not Found. A READ or WRITE imperative macro, other than WRITE AFTER(,EOF) has an ID or KEY which cannot be found within the limits of the command.

cont.

BYTE	BIT	DESCRIPTION
1	5	End of File. A READ macro has processed a record with a data length of zero.
1	6	Not Used.
1	7	Invalid ID 1. An absolute address has been specified with a record number beyond the capacity of a track. 2. An address beyond the limits of this file has been specified, either absolute or relative.

ERROR BYTES - STATUS CODE

Note 1 - $BKSZ \neq (\text{key length (if } KLEN \neq \emptyset) \text{ plus data length or data length only if in only key processing of the count section of the disc record}$

1. WRITE macros - the smaller of the two factors will determine the number of bytes transferred from memory to the disc.
 - A. If BKSZ is smaller, the balance of the disc record will be cleared to binary \emptyset .
 - B. If BKSZ is larger, the recording will be truncated.
2. READ macro - the smaller of the two factors will determine the number of bytes transferred to memory.

Note 2 - $RCSZ + KLEN \neq \text{key length (if } KLEN \neq \emptyset) \text{ plus data length of the count section of the disc record.}$

WRITE macros - see Note 1 on WRITE macros substituting $RCSZ + KLEN$ for BKSZ.

ERROR BYTES - HEX REPRESENTATION

0000	-	No Error
0001	1/7*	Invalid ID
0002	1/6	Not Used
0004	1/5*	End-of-File
0008	1/4*	Not Found
0010	1/3*	Data Check
0020	1/2	Not Used
0040	1/1	Not Used
0080	1/0	Not Used
0100	0/7	Not Used
0200	0/6	Not Used
0400	0/5	Not Used
0800	0/4*	No Room
1000	0/3	Not Used
2000	0/2	Not Used
4000	0/1*	Length Error
8000	0/0	Not Used

TYPE OF ACCESS ③	SKAD CONTENTS		
	REL=R ①	REL=T	ABSOLUTE ②
ID	YES	YES	YES
(G)KEY	NO	YES	YES
AFTER ④	NO	YES	YES

ACCESS/ADDRESS COMPATIBILITY

- ① RCFM=UNDEF not allowed
- ② Not REL=R or REL=T
- ③ Positional Parameter of Imperative
- ④ Also AFTER, EOF and RZERO

	KLEN	AFTR	RDID or WRID	IOAL CONTENTS
AFTR=YES	=∅ >2	=YES =YES	- -	COUNT, DATA COUNT, KEY, DATA
RDID=YES or WRID=YES	=∅ =∅ >2 >2	- =YES - =YES	=YES =YES =YES =YES	DATA COUNT, DATA KEY, DATA COUNT, KEY, DATA
RDKY=YES or WRKY=YES	>2 >2 >2 >2	- =YES - =YES	- - =YES =YES	DATA COUNT, KEY, DATA KEY, DATA COUNT, KEY, DATA

IOAL CONTENTS

Notes on IOAl Organization

Although a file can only be addressed one way, relative track, relative record or absolute, it can be accessed in all ways.

For example, if the specifications have been made in DTFDH; AFTR=YES, RDID=YES, WRKY=YES (or any other combinations of AFTR, ID and KEY keywords) then the breakdown of IOAl is:

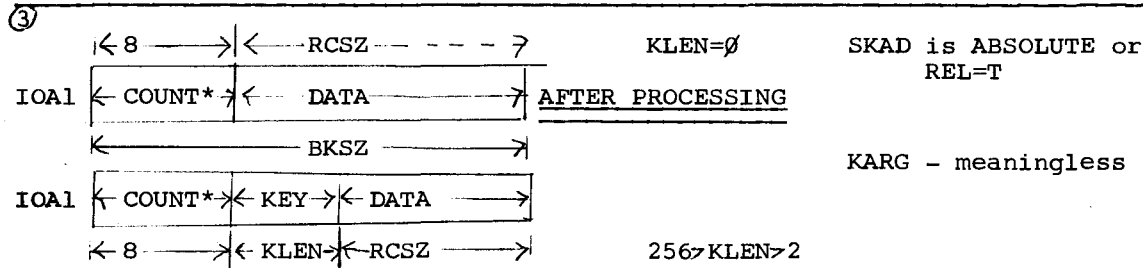
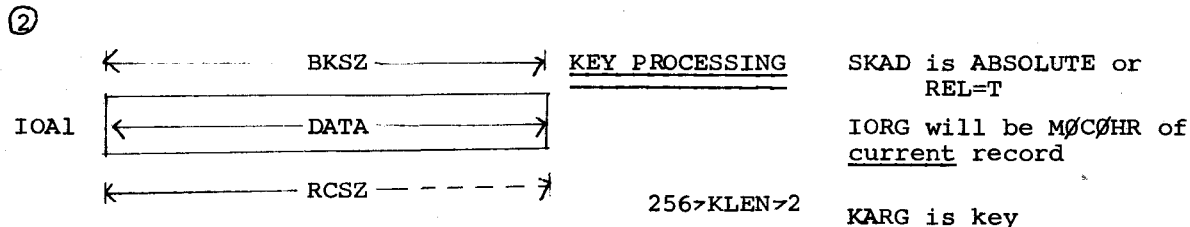
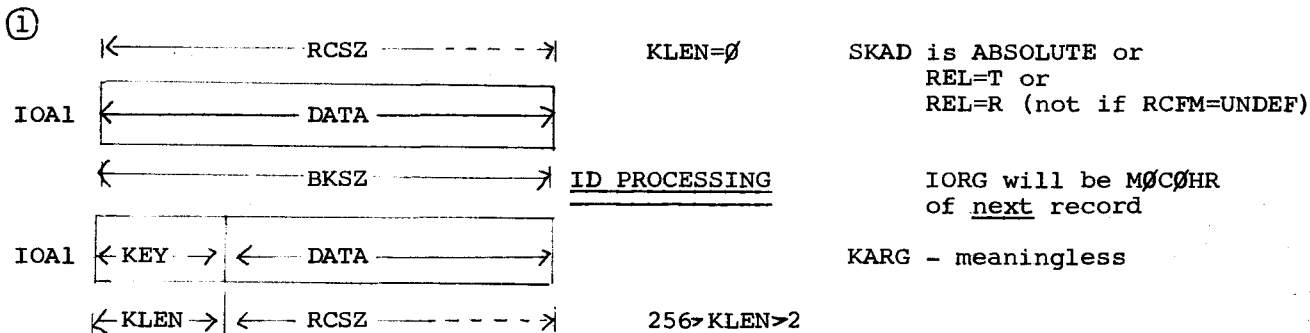
1. The first 8 bytes are reserved for the count.
2. The next KLEN bytes are reserved for the key.
3. The balance of IOAl (BKSZ-(KLEN+8)) is reserved for the data section.

That is, a READ KEY imperative macro would always supply the data of the record found at IOAl+8+KLEN and a WRITE KEY command would start recording on disc with data being supplied starting at IOAl+8+KLEN.

READ ID (and WRITE ID) would process data starting at IOAl+8 and the first KLEN bytes processed would be the key of the record.

Format write IOAl thru IOAl+7 will contain the actual disc address at the conclusion of the imperative macro.

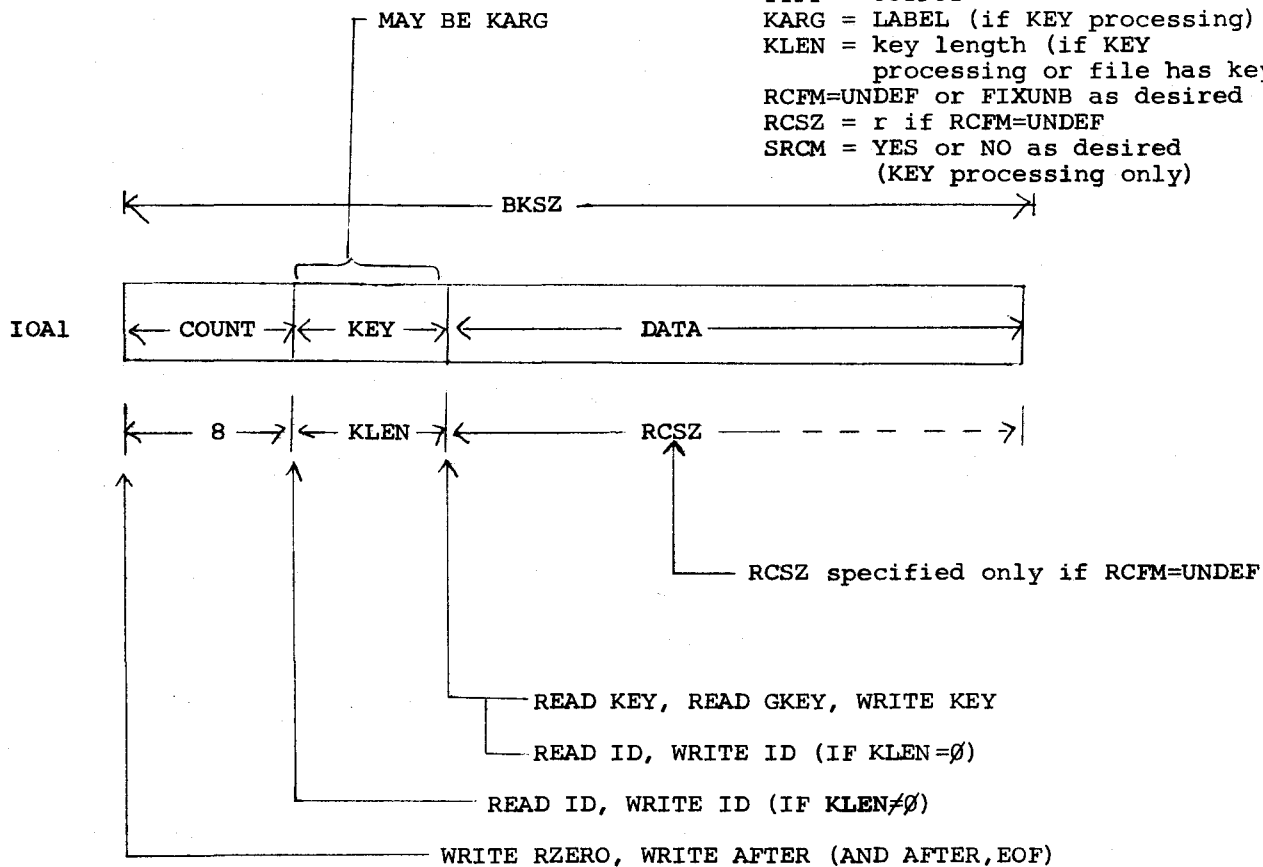
IOAl ORGANIZATION



* COUNT is indeterminate before executing.
COUNT is ØCØHRKDD after execution.

IOAI ORGANIZATION FOR COMBINED AFTER(RZERO), ID AND/OR KEY PROCESSING

SKAD is ABSOLUTE or REL=T
 IORG = LABEL as desired
 TYPF = OUTPUT
 KARG = LABEL (if KEY processing)
 KLEN = key length (if KEY processing or file has keys)
 RCFM=UNDEF or FIXUNB as desired
 RCSZ = r if RCFM=UNDEF
 SRCM = YES or NO as desired (KEY processing only)



FUNCTION	NORMAL RETURN		ERRO RETURN	
	FIND OTHER THAN LAST	FIND LAST IN FILE	NO FIND*	EOF*
ID	ID OF <u>NEXT</u> RECORD IN FILE **	ALL 1 BITS	ALL 1 BITS	ID OF EOF RECORD
KEY -NOT MULTITRACK- See SRCM	ID OF <u>THIS</u> RECORD	IF OF <u>THIS</u> RECORD	ID OF <u>NEXT</u> RECORD IN FILE **	ID OF EOF RECORD
KEY -MULTITRACK- See SRCM	ID OF <u>THIS</u> RECORD	ID OF <u>THIS</u> RECORD	ID OF <u>NEXT</u> RECORD IN FILE **	ID OF EOF RECORD
AFTR	ALL 1 BITS	ALL 1 BITS	ALL 1 BITS	ALL 1 BITS

IORG CONTENTS

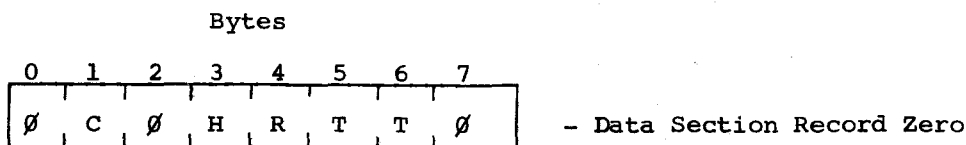
* See ERROR bytes

** All 1 bits if next record is
 beyond last extent of last
 volume of the file

RECORD ZERO (TRACK DESCRIPTOR) RECORD

In files processed with AFTER imperative macros, the data section of Record Zero of each track serves as a 'capacity record'. This capacity record is initialized on each successful WRITE RZERO macro and is checked and updated on each successful WRITE AFTER macro. It is checked only on each WRITE AFTER, EOF macro.

The format of this capacity record is:



where:

- Bytes 1 The logical cylinder no. (binary)
- 3 The logical head no. (binary)
- 4 Current last record number on this track - initially zero (binary)
- 5-6 Capacity of this track in bytes-initially.
- Space remaining expressed in bytes after each successful WRITE AFTER (binary)

Bytes 0,2,7 are reserved.

FILE CREATION (LOADING)

A. File has been pre-formatted using FOR card. See DASM.

WRITE ID

- a. SKAD may be ABSOLUTE, REL=T, or REL=R
- b. KLEN and KARG defined as desired; subsequent updating or retrieval by key? formatted with key?
- c. Supply address to SKAD (randomly developed addresses or next sequential address) and data to IOAl and issue WRITE ID.
- d. IOAl is organized as shown in the IOAl organization chart.

B. File has not been pre-formatted, or it is desired to re-establish the file.

1. WRITE RZERO

- a. SKAD may be ABSOLUTE or REL=T (record no. specified as ∅)
- b. Issue WRITE RZERO commands to all tracks of the file. This will initialize R∅ and clear the tracks, then

2. WRITE AFTER

- a. Supply address to SKAD, data to IOAl+8 and issue WRITE AFTER. Addressing may be randomly developed or sequential.
- b. IOAl is organized as shown in the IOAl organization chart.

3. WRITE AFTER, EOF

An EOF record (data length ∅ key length is KLEN) may be written after last record of a track or file.

DTFDH LAYOUT

FILENAME	
0	XO?P OPEN
4	XC?L CLOSE
8	XR?K READ KEY
12	XR?G READ GKEY
16	XW?K WRITE KEY
20	XR?Ø NOT SUPPORTED
24	XR?I READ ID
28	XW?Z WRITE RZERO
32	XW?I WRITE ID
36	XW?A WRITE AFTER
40	XW?E WRITE EOF
44	XW?T WAITF

FILETABLE		
FN + 48	XF?Ø FLID	1 [CRDT]
52	2 [XPD]T	3 ERRO
56	4 DTF FLAGS	5 BKSZ
60	6 [RCSZ] ①	7 [IORG]
64	8 ERROR BYTES	9 IOAL
68	A [KARG]	B [KLEN]
72	C [LBAD]	D SKAD
76	E XTEN	F DVNO
80	DEVICE TABLE	
84	H RECORDS PER TRK ④	I I/O FLAGS
88	J TABL ADDR	K TABL CURRENT
92	L VOL TABLE ADDR	M VOL TABLE CURRENT
96	N ADDR	LAST CAW
100	DISPATCHER	
104	P/U	CYL Re-ISSUE HEAD
108	STATUS	L/U SENSE BYTES

CCW CHAINS (MAX 1Ø)		
FN + 112	1st CCW	FT + 64
184	1Øth CCW	136

WORKING STORAGE							
FN + 192	P RØ OR UNDEF	COUNT				FT + 144	
200	Ø C	Ø H	R	KL	DL	DL	
208	Q RØ	DATA					152
216	Ø C	Ø H	R	T	T	Ø	
224	R	SKAD STORAGE					160
	S MULTI FACTOR	T DIVIDE FACTOR	U KEY GAP	V DATA GAP		168	
	W TRACK LENGTH	X HEADS	Y OVERHEAD ③	Z MAX RCRDS KEY	UNKY	176	

EXTERNAL TO DTFDH

EXTENT TABLE(S)			
EXT TYPE	EXT SEQ	LWR	EXT C
Ø	H	Ø	C
UPR	EXT H	LOG UNIT	
Ø	H	LOG UNIT	
# OF EXTS	VOL NO.	HIGH RTA	

EXTENT TABLE(S)			
EXT TYPE	EXT SEQ	LWR	EXT C
Ø	H	Ø	C
UPPR	EXT H	LOG UNIT	
Ø	H	LOG UNIT	
	VOL NO.	HIGH RTA	

VOLUME TABLE(S)		
FN + 232	VOL. NO.	EXTS IN VOL.
236	EXT. TABLE ADDR	HIGH RTA IN VOL.
24Ø	LOG UNIT	PHYS. UNIT
	VOL. NO.	EXTS IN VOL.
	EXT. TABLE ADDR	HIGH RTA IN VOL.
	LOG UNIT	PHYS. UNIT

XF?4
DTF FLAGS (FILENAME+56)

CLOSE	RCFM=UNDEF	VRFY=YES	SRCM=Ø	REL=R/T	REL=R	REL=T	TYPF=OUTPUT
WAIT =YES	IORG=LABEL	LEAD=LABEL	RDKY=Ø	RDID=Ø	WRKY=Ø	WRID=Ø	AFTR=Ø
Ø	1	2	3	4	5	6	7

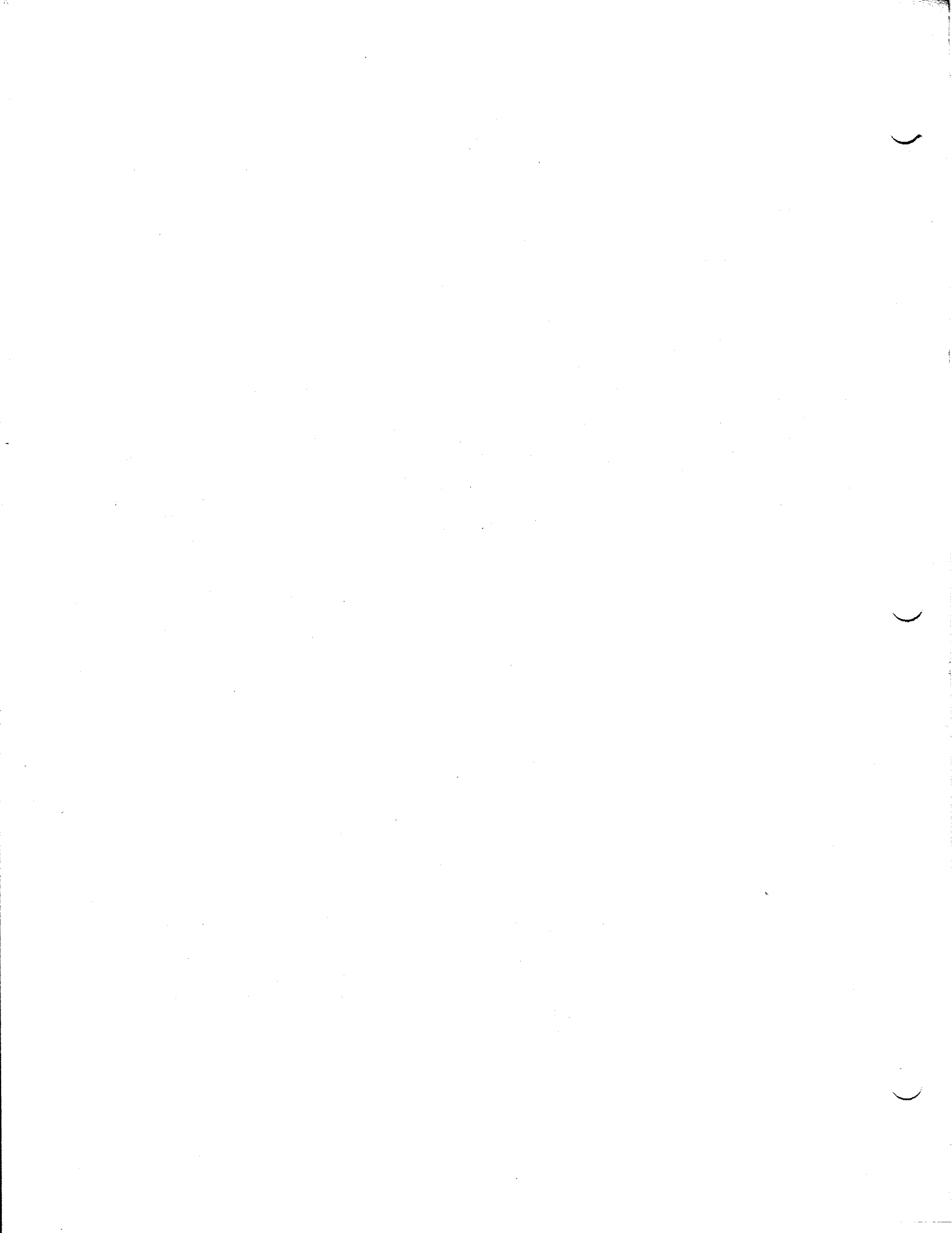
XF?I
I/O FLAGS (FILENAME+9Ø)

I/O OPEN	SEEK VOL1	FMT1	FMT3	UHL	WFML	WUHL	WUTL
Ø	1	2	3	4	5	6	7
Ø	RDKY	RDGKY	RDID	WRKY	WRID	WRRØ	WRAF
							WREF

VOLUME TABLE FLAGS (FN+233)

OPEN	CYL ALLOC ⑤	RESERVED
Ø	1	2 → 7

1. If RCFM=UNDEF, then RCSZ is DCY(r+r-16)
if RCFM=FIXUNB, then RCSZ is BKSZ+KEY GAP (-AFTER) 8
2. only in 1st of each file
3. DATA GAP+OVERHEAD
4. RCFM=FIXUNB only
5. if SRCM=YES and volume has file on cylinder allocated



APPENDIX W. DTFMT, DMTIO, DMTIT

A. DTFMT

To correct the following problem:

When the proc call line specified: TYPE=INPUT, or when not used, the default is INPUT. Some coding was not generated which could cause a PROC ABN or erroneous results.

B. DMTIO

The following problems have been corrected:

1. When the proc call line specified OUPT=NO, some coding was not generated causing an undefined (U) error in the listing.
2. The correction of a tag conflict which could result in not getting a valid 'EOV1' trailer label.
3. If the checkpoint constant and the 'EOF', 'EOV', and 'HDR' constants fell at the beginning of a memory block being dumped by the checkpoint routine, errors could occur during checkpoint restart.
4. If the keyword parameter INPT=NO was specified, undefined (U) errors occurred.
5. Redesign of the coding for 2Øu3 or 2Øu4 halts to insure the return branch is to the coding which initiated the branch to the error coding. Prior to this, the error coding always returned to the OPEN coding.

The following enhancement has been made:

A check for tape DEVICE-ID during the OPEN. If the unit is not assigned to tape, an X'600E' error message will be issued.

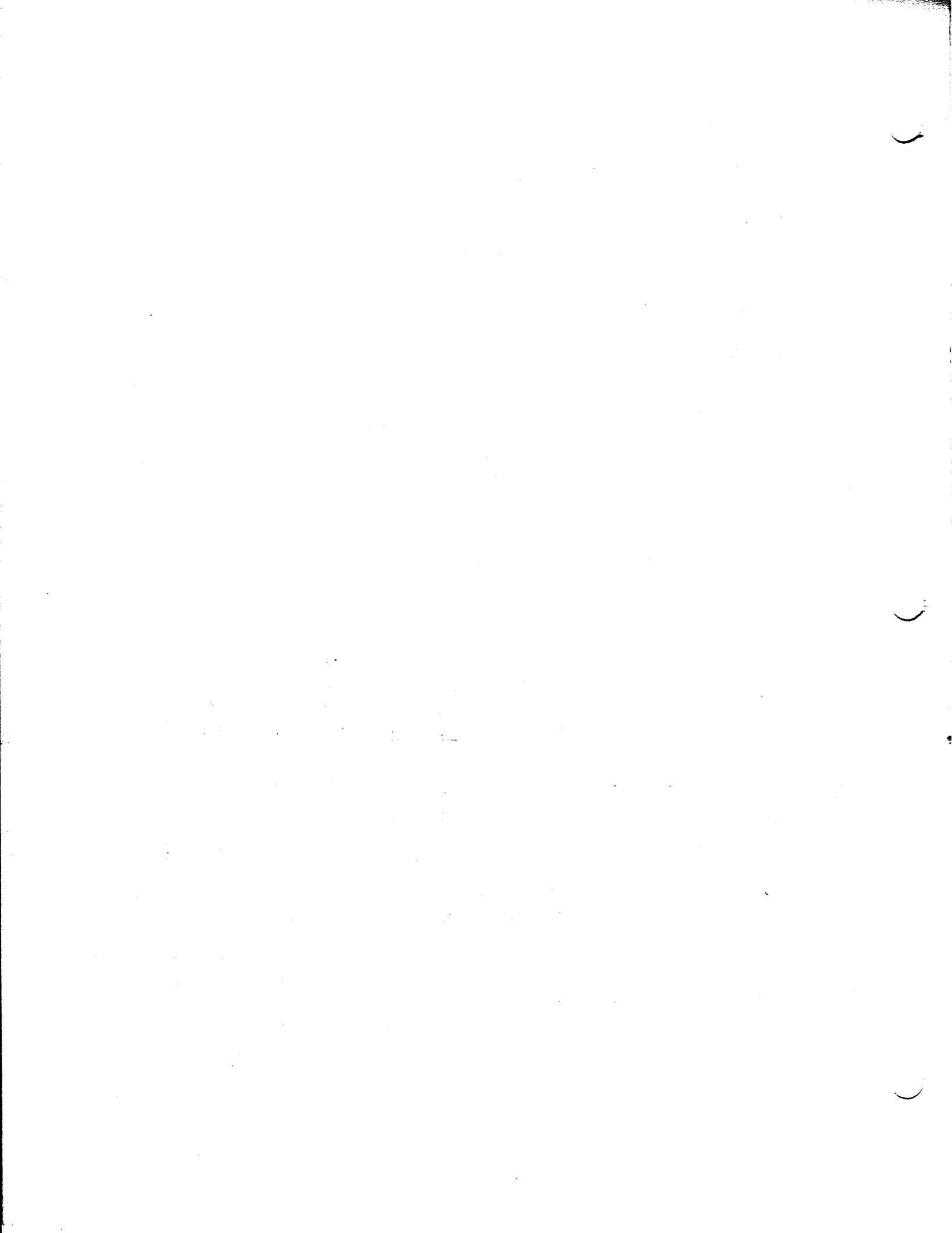
C. DMTIT

The following problems have been corrected:

1. When generating, using the following call, two undefined (U) errors were given:
`DMTIT GET=TRAN,PUT=TRAN,WORK=YES`
2. If the checkpoint constant and the 'EOF', 'EOV', and 'HDR' constants fill at the beginning of a memory block being dumped by the checkpoint routine, errors could occur during checkpoint restart.

The following enhancement has been made:

A check for tape DEVICE-ID during the OPEN. If the unit is not assigned to tape, an X'600E' error message will be issued.



APPENDIX X. THE SYSTEM EXECUTION AREA.

1. Introduction.

The System Execution Area and its related software were intended to be used in an OS-500 environment to make possible faster response to inquiries and to decrease the impact of inquiries on the main ("batch") job. However, there is nothing which prevents its use in any other disc-resident operating system environment and we highly recommend its use to any installation to whom throughput is important. While certain similarities may be observed between this 9300 software and the use of a system execution area under OS-4 or other operating systems, programmers should not make any assumptions about the 9300 software based on their knowledge of other systems.

2. Definitions.

System Execution Area - A single-extent file, with a file-i.d. of "SYSEXAR". This file must be cylinder-aligned and occupy a whole number of cylinders, up to a maximum of 64. At any given moment during the operation of the 9300 operating system, there is only one System Execution Area. It is the file labelled SYSEXAR on the disc drive from which the operating system was booted. It is to this file that programs are written by the Formattor and from which programs are loaded when the execution of programs in the system execution area is requested.

Formattor - As of this software release, a new utility called "SHEAF" (for System Highspeed Execution Area Formattor) is being supplied. The necessary statements for linking SHEAF are included with those for all other utilities in the run stream element CDS. The purpose of SHEAF is to pre-format the System Execution Area and to write loadable programs in it. It does this as directed by commands presented to it in the control stream (see "Formattor Operation" below). SHEAF's input consists of loadable programs in library files. Its output consists of more efficiently written copies of these programs in the System Execution Area. The programs are written out in blocks of sizes varying up to the maximum capacity of a track (3625 bytes for 8411, 7294 bytes for 8414). SHEAF also generates indices and other control information necessary for loading the programs and for maintaining SYSEXAR.

Fast Loader - As of this release, a transient named "FAST?LDR" is supplied which is written to head 13, record 1 of TRANSCYL by TRAN. The purpose of this transient is to locate and load programs from SYSEXAR. Once a program is loaded, the Supervisor "remembers" which loader was used to load the program and uses the same loader to load overlays of that program in response to the FETCH macro.

Slow Loader - This is the familiar Supervisor program locator/loader routine which is generated as a part of the Supervisor and written to head 0, record 1 of TRANSCYL by TRAN. This loader locates and loads programs from library files. The additional code which was required within the Supervisor for choosing between fast and slow loaders was more than offset by economies of coding achieved in other areas.

3. Advantages and Disadvantages of Using SYSEXAR.

If you have observed the operation of your system for any length of time, you have noticed the delay of several seconds which occurs at each job step when the Supervisor must locate and load first the Job Control Program, then the user program named in an EXEC statement. This delay occurs because, when the Supervisor locates each program, it must first search the Volume Table of Contents (VTOC) for the file containing the program, then search the file directory by reading one record after another until it locates the directory entry for that program. To load the program, it must read one record at a time from the data extent of the library file.

Each 160-byte record contains a maximum of 124 bytes of program text. To load a program consisting of 12.4-K bytes of text, therefore, at least 100 disc accesses must be made, each consuming at least one disc revolution. Such a program would occupy 4 8414 or 6 8411 tracks. If the same program were copied to SYSEXAR by SHEAF, it could then be located by means of a single search-on-key command and loaded by a single chain of read commands consuming only two or four disc revolutions. It would occupy 2 8414 tracks or 4 8411 tracks. It should be obvious that several seconds would be saved at each job step. The performance of a segmented program which issues numerous FETCHes would be improved noticeably.

On the debit side, programs which are to be loaded by the fast loader must always be located on the system resident disc unit. Their maintenance may not be as easy as the maintenance of a standard library file by LIBS since SHEAF does not offer any means of copying a system execution area (or portions of it) from pack to pack. SHEAF also does not have any ability to compress out programs which have been deleted. When SYSEXAR is full, it must either be rebuilt by a new SHEAF run to copy all desired elements from library files, or must be expanded by using DASM to create a larger SYSEXAR, perhaps at the expense of other files on the same pack. Although SYSEXAR may be copied to tape or to a backup pack by the Dump/Restore Utility (DPRS), this procedure still does not condense the amount of material contained in the file. In addition, should SYSEXAR be destroyed through hardware, software or other error, it may be necessary to have the original library-format loadable programs available so that it can be rebuilt. This could necessitate tying up library file space.

You should weigh these pros and cons when deciding which programs should be placed in SYSEXAR. OS-500 installations will obviously prefer to put all, or most, of their inquiry programs in SYSEXAR. Job Control should also be included. Other candidates for inclusion in SYSEXAR would be any user programs or UNIVAC utilities which are used frequently or any program which makes significant use of segmentation. If Job Control is placed in SYSEXAR, a copy of JBCN must be preserved in SYSFILE, since it is this JBCN that is loaded at IPL time. It would not be desirable to put TRAN in SYSEXAR since TRAN must be capable of running in the absence of any transients and the fast loader transient must be called to load all programs from the System Execution Area. It may not be desirable to put SHEAF in SYSEXAR because of the possibility of it being destroyed by errors occurring during its own execution. The RPG compiler (R11T) must not be put in SYSEXAR since its operation is dependent on its ability to interface with the "slow" locator/loader routine.

4. How to Execute Programs in SYSEXAR.

A program which has been copied to SYSEXAR by SHEAF may be executed through the control stream by means of the following statement:

```
/ EXEC program-name,EX
```

The operation of a program which is executed from SYSEXAR is indistinguishable with that of a program which is loaded from a conventional library file. Job Control is only able to determine, when loading a program from the System Execution Area, whether or not it requires the use of tape units. There is no information as to which tape units are needed. Programs which are loaded from the System Execution Area may be used in conjunction with the Transient Tape Dispatcher in systems generated with that option.

5. Formatter Operation.

The System Highspeed Execution Area Formatter (SHEAF) is loaded and executed by the following control stream:


```

/      EXEC SHEAF
/      DATA T
.
. commands
.
/*

```

The Formatter recognizes three commands - FORMAT, ADD and DELETE, whose use is described below. The commands are free-form to the extent that they may begin in any column from 1 through 16. Operands, if any, must be separated from commands by at least one space and must end before column 73. Spaces may not be embedded in commands or in operand strings. SHEAF generates no halts other than usual IOCS displays. It produces a number of self-explanatory error messages. All error messages are preceded by "!" in print positions one through eight. Informational messages are preceded by "*" and input statements by "\$" in the same positions. Any card whose first sixteen columns are blank is ignored and not printed.

a. The FORMAT Command.

The FORMAT command instructs SHEAF that it is to initialize the System Execution Area. This command, if used, must be the first statement read by the Formatter; it will be ignored if it is preceded by any other command. It has a single, optional parameter, a decimal number which defines the number of tracks which are to be pre-formatted for index use. If the parameter is omitted, SHEAF formats a single index track. The remaining tracks of SYSEXAR are erased.

Examples:

FORMAT 5 causes formatting of 5 index tracks

FORMAT (blanks) causes formatting one index track

Each index track holds 17 records if SYSEXAR is on an 8411 disc, 27 if 8414. The first record of the first index track is an "index control block" and is used by SHEAF for maintenance purposes. This index control block is updated after successful completion of all processing associated with each command. If any situation occurs during the processing of a command which could cause invalid information to be written on the index control block or invalid program text to be copied, SHEAF cancels the job. The contents of SYSEXAR, its index, and the index control block are usable through the last successfully-processed command.

Each index record consists of a 16-byte key containing program-i.d. and phase-i.d. and a 104-byte data area which includes such control information as program size, base address and transfer address, as well as the command control word (CCW) chain which the fast loader will use to load the program. All information in the index record which pertains to disc locations is made relative so that SYSEXAR may be copied to any disc location. The address fields of transfer-in-channel (TIC) and search-i.d. commands are also relocatable and are converted by the fast loader transient to absolute addresses at load time. If a single-phase program is written without the "NOCLEAR" option (see "ADD Command" below), it requires only one index record. The number of index records required by programs written with "NOCLEAR" specified depends on the degree of fragmentation of the text of each program.

When SHEAF adds so many programs to SYSEXAR that it exhausts available index space, it makes use of the first track on the next available cylinder as an index track and pre-formats it before writing any index information on it. If you anticipate including a large number of programs in your System Execution Area, it is to your advantage to

allocate, via the FORMAT statement, several index tracks. This will cause faster access time for any program in SYSEXAR. We have found that about one index track per 15-20 text tracks will be necessary. The disc allocation map at the end of each SHEAF listing may be helpful in judging the relative amount of index space required for your programs.

The FORMAT command is only used when it is necessary to initialize or rebuild SYSEXAR. If SHEAF is being used for adding programs to an existing SXSEXAR, FORMAT must not be used.

b. The ADD Command.

The ADD command instructs SHEAF to search for a loadable program in a library file and then to copy it to SYSEXAR. If the program is found, any program which previously existed in SYSEXAR which had the same name is deleted. Any index records which are required for locating and loading the program being ADDED are written. The ADD statement has four parameters which must be separated from each other by commas; omitted parameters must be accounted for by consecutive commas but trailing commas may be omitted. The four parameters are:

1. program-i.d. - the one to eight-character name by which the program is located and loaded. This parameter is required in each ADD command.
2. logical unit number - a one or two-digit hexadecimal value; the disc unit containing the library file which contains the program to be ADDED. This parameter is optional; until some logical unit number is specified, zero is assumed; if an ADD command omits logical unit number, the last number specified is assumed.
3. file-i.d. - the name of the library file containing the program to be ADDED. This parameter is also optional; until some file-i.d. is specified, SYSFILE is assumed; whenever it is omitted, the last file-i.d. specified is used.
4. mode of output - NOCLEAR or omitted; if this parameter is omitted, SHEAF assumes that the entire area occupied by the program, as defined by its program-i.d. record, is to be cleared to binary zeros where no text is provided and that the program is to be written in SYSEXAR with the largest blocksize possible. If NOCLEAR is specified, only those areas which are to contain program text will be written out. This may result in slower loading and less efficient utilization of disc space but there are situations where the NOCLEAR option must be specified.

Examples:

```
ADD JBCN (1)
ADD LIBS (2)
ADD KSMB (3)
ADD INQ1,,LOADLIB (4)
ADD REPORT5,1,RPGOUTPT,NOCLEAR (5)
ADD INQ2,2,LINKFILE (6)
ADD INQ3,,TESTFILE (7)
ADD X,Ø1 (8)
```

Examples (1) through (3) above add programs from SYSFILE on logical unit zero.

Example (4) adds a program from LOADLIB on logical unit zero.

Example (5) adds an RPG-generated program.

Example (6) adds adds a program from LINKFILE on logical unit two.

Example (7) adds a program from TESTFILE on logical unit two.

Example (8) adds a program from TESTFILE on logical unit one.

Use of the NOCLEAR specification is mandatory in the following cases:

- (1) If a program contains patches introduced by LIBS via the DATA statement, and if these patches include branches to locations outside the area occupied by the program.
- (2) If a program contains ORG statements adjusting the location counter to absolute addresses outside the program area.
- (3) If some area within the program is to be used for receiving information from a previously-executed program. Areas defined by the COM Assembler directive or COMMON statement in FORTRAN are examples of such communications areas.

If a program includes large buffer areas which do not need to be pre-cleared and its text (instructions and constants) is contiguous (not broken up by DS or ORG statements), a more efficient use of SYSEXAR will result from specifying NOCLEAR. The loadable programs produced by the RPG compiler are usually so contiguous that their copying with NOCLEAR specified may save some space in SYSEXAR.

When SHEAF copies a segmented program and NOCLEAR is not specified, it will write out the root phase such that the whole area occupied by the program will be cleared when the root phase is loaded. It writes out all overlays as though NOCLEAR were specified, thus causing any work areas to be preserved when overlays are loaded.

c. The DELETE Command

The DELETE command instructs SHEAF that it is to delete a program from the System Execution Area index. The key portion of any index record(s) pertaining to the program is filled with 16 bytes of X'01'. The DELETE command uses one mandatory operand, which is the name of the program to be deleted.

Examples:

```
DELETE ABC
DELETE /XYZ
```

Deleting programs from SYSEXAR does not cause any freeing of space or compression to take place; it only makes the named program unlocatable. It is not necessary to delete programs before adding new programs with the same names; this is accomplished as a part of the ADD processing. Only one copy of a program with a given program-i.d. can exist in SYSEXAR at one time.

d. Miscellaneous Features.

When SHEAF is executed, it always updates the location within the Supervisor which contains the starting and ending cylinder numbers of SYSEXAR. If you have reallocated files so as to cause SYSEXAR to have different boundaries, you must either execute SHEAF or reboot the system to cause this information to be brought up to date.

If SYSEXAR has been expanded, whether or not it starts in the same place, SHEAF is aware of this and erases the additional tracks (providing it is able to read an index control block) before processing any commands.

If it is only necessary to obtain a listing, or map, of the contents of SYSEXAR, SHEAF may be executed without any input statements between the / DATA T and /* cards. The map always includes the location of the next available text and index records; this gives an indication of how close to being full SYSEXAR is.

In DELETE and ADD commands, program-i.d.'s may optionally be preceded by a slash. There is no capability for searching within library groups; the first loadable program with the name specified will be copied.

APPENDIX Y. HALTS.

The following are new or changed halt displays pertinent to this release. This information should be inserted in your copy of Halt Displays Programmer/Operator Reference (UP-7719 Rev 4).

<u>HALT</u>	<u>SOFTWARE AREA</u>	<u>MEANING</u>	<u>RECOVERY ACTION</u>
ØBxx	RPG	Each bit of xx display corresponds to a H1 thru H8 halt indicator. X'Ø1' represents H8 X'Ø2' represents H7 X'Ø4' represents H6 X'Ø8' represents H5 X'1Ø' represents H4 X'2Ø' represents H3 X'4Ø' represents H2 X'8Ø' represents H1 If more than one bit is set, the corresponding H1 thru H8 RPG halt combinations have occurred.	To continue, depress START switch.
ØEEE	RPG	An EXCPT calculation specification and at least one output specification (containing an E in column 15) were not present. Both specifications are required.	Press START to cancel the job. Correct the program and restart.
23uØ	Sequential Access Method (SAM)	A file on physical unit u has been defined as an add file but its Format 1 label next-record pointers contain binary zeroes.	To retry the read of the Format 1 label, press START. To cancel the job, reply Ø1. To start writing data at the beginning of the file, reply Ø2 or greater.
23uD	Sequential Access Method (SAM)	Creation/expiration date error on unit u.	To retry the Format 1 read, press START. To cancel the job, reply Ø1. To ignore the error, reply Ø2 or greater.
24xx	Direct Access Method (DAM)	See Appendix V for meaning and recovery action for all 24xx halts.	
2Bh1	Dump/Restore (DPRS)	The unit type in column 7 of card h is not appropriate for the specified function; OR a dump to or restore from cards was requested but either no punch IOCS was linked to DPRS or insufficient memory exists to carry out such functions.	See "2Bhe" in halt manual for recovery.

<u>HALT</u>	<u>SOFTWARE AREA</u>	<u>MEANING</u>	<u>RECOVERY ACTION</u>
40uE	Fast Locator/ Loader	In addition to other meanings, this halt may indicate that SYSEXAR has not been allocated or that its Index Control Block is unreadable.	None. Press START to cancel the job.
4155	Job Control	The program to be executed is linked so low as to overlay the Supervisor.	None. Press START to cancel the job.
4171	Inquiry IOCS (DTFUQ)	In addition to other meanings, this halt may indicate that this generation of DTFUQ is incompatible with the type of inquiry device or that a single-inquiry-mode program is being executed with a Supervisor that was generated without the SQWS parameter.	None. IOCS goes to Supervisor end-of inquiry routine.
41A3	Concurrent Supervisor	The buffered print file could not be located.	When START is pressed, Supervisor locator MSG 14xx or 40xx is displayed. To this second message, reply non-zero to cancel or zero to retry the search.
41C0	Job Control	TRANSCYL has not been initialized for this Supervisor.	Press START to continue. This halt is to be expected when booting from a newly-created system pack; at any other time it would indicate that the contents of the transient file have been destroyed. TRAN must be run immediately to initialize TRANSCYL.
41C1	Job Control	The disc pack from which the system was booted does not contain a file labelled TRANSCYL.	None. This system pack is unusable until it has had TRANSCYL allocated on it by the Direct Access Space Management program (DASM).
41C2	OS-500 Job Control	Unable to format the roll-out file due to disc errors.	Press START to retry. If halt persists, area occupied by ROLLOUT must be reprepped.

<u>HALT</u>	<u>SOFTWARE AREA</u>	<u>MEANING</u>	<u>RECOVERY ACTION</u>
41C3	OS-500 Job Control	The roll-out file is not cylinder-aligned.	Press START to continue processing. It will not be possible to activate non-resident inquiry programs until ROLLOUT is re-allocated on cylinder boundaries.
41C4	OS-500 Job Control	The roll-out file is not big enough to roll out the largest possible main job to be run in this system.	As for 41C3. ROLLOUT must be reallocated.
41CA	Supervisor	In a PARAM card, an expected comma was not encountered or a PARAM R card did not end on a blank column.	Clear the reader, correct the erroneous PARAM card, and press START to retry.
41CB	Supervisor	The address in a PARAM card would cause the patch to be applied to restricted memory, to areas beyond the size of the machine, or (if a PARAM R card) the address is odd.	Same as 41CA.
41CC	Supervisor	In a PARAM card, the digits making up the address, length, or (PARAM R only) contents of a patch were not hexadecimal digits (from zero through F).	Same as 41CA.
41Du	Job Control	The volume serial number of the disc on <u>physical unit</u> u does not match the one specified in a VOL statement.	Mount the correct disc pack on <u>physical unit</u> u, then press START. To cancel, make an X'FØ' operator request, then press START.
41Eu	Job Control	In processing a VOL statement, Job Control was unable to read the VOLl label on <u>physical unit</u> u. Either a read error occurred OR the record read was not a VOLl label.	Same as 41Du; it may be necessary to reprep the disc pack.

<u>HALT</u>	<u>SOFTWARE AREA</u>	<u>MEANING</u>	<u>RECOVERY ACTION</u>
44us	Fast Locator/ Loader	While trying to locate or load a program in SYSEXAR on physical unit u, an error status occurred. Possible values of s are: 1=software error 2=data check 3=end of file 4=no-find 8=wrong-length	Press START to cancel; make non-zero reply to retry.
69u4	Supervisor (disc dispatcher)	Track overrun has occurred on physical unit u.	Press START to skip the disc request packet. The track on which the error occurred must be erased or reprepmed.
69u5	Supervisor (disc dispatcher)	An invalid command sequence was issued to physical unit u.	Press START to skip the disc request packet. A program error may have caused the CCW chain to contain an invalid command or otherwise issue a command that could not be accepted by the disc control unit.
6EEE	Sort/Merge (PHASEZERO)	Number of records written by the Sort does not equal number of records read.	Reply X'Ø1' to ignore the error; any other reply causes job cancellation.
74BB	Transient write-out (TRAN)	Insufficient room remains on the current track to write out the requested transient.	Same as for 7444.
74EE	Transient write-out (TRAN)	TRANSCYL cannot be located on the output unit.	Press START to retry the search.
74FF	TRAN	The relocatable module which was to be written out as a transient cannot be located on the input unit.	Same as for 74EE.

P. O. BOX 500
BLUE BELL, PENNSYLVANIA 19422
TELEPHONE (215) 542-4011

SOFTWARE ORDER SERVICES
(215) 542-3421

SOFTWARE SYSTEM SUPPORT REQUEST
DIV. MKT. ORGN. ASSIGN.

<u>SOFTWARE REQUESTED</u>	<u>MEMO #</u>	<u>DATE</u>	<u>MEDIA REQUIRED</u>
<input type="checkbox"/> 1. 8411/14 DNCOS (Via Tape) Rev. 5.0.0	147	4-75	2 - Tapes
<input checked="" type="checkbox"/> 2. 8411 DNCOS (Disc) Rev. 5.0.0	"	"	2 - Disc/Tapes
<input type="checkbox"/> 3. 8414 DNCOS (Disc) Rev. 5.0.0	"	"	1 - Disc/ 2 - Tapes
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

** CHECK BOX FOR SOFTWARE REQUIRED.

REQUESTED BY:

ATTN: _____

SIGNATURE APPROVED
USER D. P. MGR/LOCAL S. A. MGR.

SPECIAL INSTRUCTIONS:

SEE PAGE 3 OF 3 - ORDERING PROCEDURES

REMARKS AND/OR ADDRESS CORRECTION

SEND REQUEST FORM TO:
SPERRY UNIVAC
SOFTWARE ORDER SERVICES
TOWNSHIP LINE AND JOLLY RD.
BLUE BELL, PA. 19422

NOTE: ALL USERS OUTSIDE THE UNITED STATES SHOULD CHECK WITH THEIR LOCAL UNIVAC SUBSIDIARY OFFICE FOR DISTRIBUTION PROCEDURES.