AD-754 123

USER'S GUIDE TO THE 160A AND 1604A UTILITY
PROGRAMS

Fleet Operations Control Center
Pacific Fleet
FPO San Francisco 96617, California

10 January 1973

AD754123

USER'S GUIDE TO THE 160A AND 1604A

UTILITY PROGRAMS

PROJECT NO. F12S280

(SUPERSEDES FOCCPAC DOCUMENT TN-5)

DDC

JAN 24 1973

FLEET OPERATIONS CONTROL CENTER
U. S. PACIFIC FLEET
FPO SAN FRANCISCO 96617

FOCCPAC:C3A2:na
5231
Ser 88
10 January 1973

From: Commanding Officer, Fleet Operations Control Center,
U. S. Pacific Fleet
To: Distribution List

Subj: User's Guide to the 160A and 1604A Utility Programs; forwarding of

Ref: (a) FOCCPAC Document No. TN-5

Encl: (1) FOCCPAC Document No. F12S280 TN-12

1. Enclosure (1), which supersedes and cancels reference (a), is
forwarded for information and retention.

2. Enclosure (1) contains the complete documentation available for the
160A and 1604A Utilities available at FOCCPAC. Requests for further
information or assistance should be addressed to the FOCCPAC Executive
Systems Group, C331, at (43) 14556.

J. I. BRADBURY
By direction

Distribution:
CINCPAC (J02C)   5
CINCPACFLT (Code 99)   5
NAVCOSSACT PACOMDET   5
NAVCOSSACT WASH D. C.   5
LANTCOMOPCONCEN   5
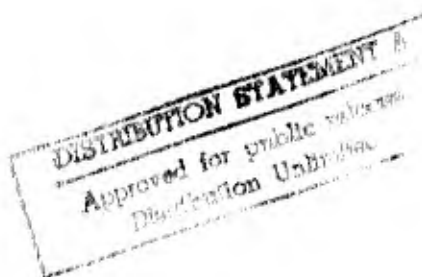DDC, CAMERON STATION, VA. 10
LIBRARY   20

1a

FLEET OPERATIONS CONTROL CENTER
U. S. PACIFIC FLEET

User's Guide to the 160A and 1604A
Utility Programs

(Supersedes FOCCPAC Document No. TN-5)

Technical Note

10 January 1973
FOCCPAC DOCUMENT NO.
F12S280
TN-12

1 b

FOREWORD

This manual includes documentation for utility programs for the 160A and the 1604A computers. All available programs for the FOCCPAC system are included, whether derived locally or from NAVCOSSACT. No distinction is made as to the source of the program. All utilities are controlled through the FOCCPAC Executive Systems Group. If any problems arise, the user should notify the Executive Systems Group, Code C331 at 14556.

The user is reminded that while nonsystem utilities are available, he should avoid using them whenever possible. If a nonsystem utility is desired, it is suggested the user contact the Executive Systems Group to see if an alternative is available.

RECORD OF CHANGES

CHANGE NO.	DATE OF CHANGE	ENTERED BY

TABLE OF CONTENTS

SECTION IV   1604A NONSYSTEM UTILITIES

None available

APPENDICES
A. Summary of 160A Nonsystem Utility Programs
B. Summary of 1604A System Utility Programs

SECTION I

160A SYSTEM UTILITIES

1-101

SECTION I

160A SYSTEM UTILITIES

1.1   DOCUMENTATION

Documentation for the 160A System Utilities is contained in FOCCPAC
Document No. F12S122 TR-51, "ACE - AUTOMATED CONTROL ENVIRONMENT."

1.2   PROGRAMS

System Utilities are implemented on the CDC 160A computer under the
following names:

        1) ASP (Automated System Program)
        2) I/O (Input/Output control program)
        3) DFM (Disk File Manager)
        4) D-160 (Display system driver/monitor)
        5) TRIADS (Transmit, Receive, Interrogate, And Display
System - remote display system driver/monitor)

SECTION II

160A NONSYSTEM UTILITIES

2-1.1

SECTION II

160A NONSYSTEM UTILITIES

2.1   NONSYSTEMS UTILITY BOOTSTRAP AND THE NONSYSTEM LOADER
       SEGMENT (FILE SEGMENTS 0000 AND 0067)

2.1.1   Function

To call all 160A nonsystem utilities and maintenance segments from the
program module of the ACE disk system or magnetic tape.

2.1.2   Equipment

NSU bootstrap tape and access to the ACE computer.

2.1.3   Options

Load from disk or magnetic tape. Details below.

2.1.4   Jump Keys/Stop Keys: None.

2.1.5   Operating Instructions

   a. Master clear.
   b. Load the N/S bootstrap (mylar tape) program.
   c. Programmed halt P=0370   A=4040
       Good halt; if any difference do steps a and b again.
   d. Master clear.
   e. Reset bank to the bank that you bootstrapped in.
   f. Hit run.
       Programmed halt   P=6000   A=0000
   g. Enter desired segment number in the A-register.
   h. Hit run.
       Programmed halt   P=6111   A=0000 Good stop
                                      A=0005 Seg not found
                                      A=0010 Seg BLK CKSUM Error
                                      A=0015 LD BLK CKSUM Error
                                      A=0020 INFO BLK CKSUM Error
                                      A=0025 MISPLACED EOF
                                      A=0030 PARITY Error
                                      A=0035 Control Abort (5 tries)
                                      A=0040 Undetermined status

   FOR TAPE OPTION:
   NOTE: To load from tape do steps a through f as above.

   i. Master clear.
   j. Set the ENTER Key up and set the P-register to 6004
       and enter the WXYZ of the tape drive in the Z-register,
       step the computer.

k. Master clear, set the P-register to 6000 and the
    desired segment number in the A-register. Reset the
    ENTER Key to neutral position.

l. Hit run.
    Programmed halt   P=6111   A=0000 Good halt
                                     A=0005 Segment not found
                                     A=0010 Seg BLK CKSUM Error
                                     A=0015 LD BLK CKSUM Error
                                     A=0020 INFO BLK CKSUM Error
                                     A=0025 MISPLACED EOF
                                     A=0030 Parity Error
                                     A=0035 Control Abort (5 tries)
                                     A=0040 Undetermined status

2.2   BI-OCTAL PAPER TAPE PUNCH (FILE SEGMENT 0001)

2.2.1   Function

This routine will dump memory in octal machine load-format on seven
level paper tape.

NOTE: Machine load-format specifies that there will be a seventh level
       punch in the first and each alternating frame. A seventh level
       punch indicates the upper half of a 160 word. No seventh level
       punch indicates the lower half of a 160 word.

2.2.2   Equipment

Console paper tape punch.

2.2.3   Options: None.

2.2.4   Jump Keys/Stop Keys: None.

2.2.5   Operating Instructions

a. Load segment 0001 from disk in any bank other than the
    bank you want to punch.
b. Master clear.
c. Enter the bank to be punched in the A-register.
d. Hit run.
    Programmed halt   P=0127   A=0000
    Set A to the first word address of core to be punched.
e. Hit run.
    Programmed halt   P=0133   A=0000
    Set A to the last word address + 1 of core to be punched.
f. Turn on console paper tape punch and hit run.
g. Final halt P=0171   A=CK sum value.*

*NOTE: The newly punched paper tape should be loaded so that the value
        in the A-register can be compared against the CK sum to determine
        if the paper tape is good. The value in the A-register must equal
        CK sum.

2.2.6   Messages/Errors: None.

2.3   AUTODIN CODED CARDS TO OPCON CODED CARDS (FILE SEGMENT 0004)

2.3.1   Function

This routine will convert AUTODIN coded cards to OPCON coded cards so that they can be prestored for use within the FOCCPAC system.

2.3.2   Equipment

Card reader and card punch on internal channel.

2.3.3   Options: None.

2.3.4   Jump Keys/Stop Keys: None.

2.3.5   Operating Instructions

    a.   Load segment 0004.
    b.   Master clear.
    c.   Hit run.

2.3.6   Messages/Errors

a.	A-REG = 2500	Could not locate a card reader, ready a reader and hit run.
b.	A-REG = STATUS	Reader error. Set P-register to 0322, correct error and hit run.
c.	A-REG = 1500	Could not locate a card punch. Ready a card punch and hit run.
d.	A-REG = 0001	Punch error or punch stop. Clear punch and hit run.
e.	A-REG = 1111	Good end of job.

If another run is necessary, hit run to continue.

2.4   PAPER TAPE TO MAGNETIC TAPE OR OPEN-ENDED FILE (FILE SEG-MENT 0005)

2.4.1   Function

This routine will prestore any 5 level paper tape to magnetic tape or an open-ended file. Magnetic tape output is foreign tape. It will also prestore OPCON or AUTODIN cards to an open-ended file.

2.4.2   Equipment

Paper tape reader, typewriter, magnetic tape drive, card reader and printer.

2.4.3   <u>Options</u>

     a. Paper tape to magnetic tape prestore.
     b. Paper tape to open-ended file prestore.
     c. OPCON or AUTODIN cards to open-ended file prestore.

2.4.4   <u>Jump Keys/Stop Keys</u>

     a. Jump Key 4 - Bypasses the typewriter.
     b. Jump Key 1 - Sequences input in columns 75-80 in increments of 10.
     c. Jump Key 2 - Prints data being prestored.

NOTE: If a printer is to be used, Jump Key 2 must be set at job start time. The Jump Key may then be lowered at any time during the prestore, and printing will be suppressed until it is elevated again.

2.4.5   <u>Operating Instructions</u>

     a. Load segment 0005 into bank 1.
     b. Master clear (set Jump Key 2 if printing is desired) and hit run.
     c. Typewriter will output "PRESTORE ID=". Type in the file ID of an open-ended file, the word "TAPE" if prestoring to magnetic tape, or "CARD" if prestoring from card to open-ended file. If an error is made, type / (slash) to delete then retype correctly. To terminate typing, type a period.

2.4.6   <u>Messages/Errors</u>

     a. PRESTORE ID= - Type in the PRESTORE ID of an open-ended file, "TAPE" if prestore is paper tape to magnetic tape, or "CARD" if prestore is from OPCON or AUTODIN cards to an open-ended file.

     b. WXY= - Type in WXY of tape unit. This type-out occurs when prestoring to magnetic tape.

     c. READER NOT READY - The paper tape reader is not ready. Make reader ready and type a space. If no more paper tape input is to be processed, hit the carriage return. This will cause the computer to close the output file and halt at address 100. To continue with next task, hit run.

     d. NO STATION FOUND IN ALERT STATUS - No paper tape stations were found alert and ready. Alert input station and type a space.

e. ILLEGAL - The input parameter was illegal. Retype to continue.

f. BUS PARITY ERROR FROM PT STATION - A bus parity error was detected in the PTCU. Restart program.

g. READER NOT IN FIVE LEVEL MODE - The paper tape station is not in five level mode, switch to five level and type a space.

h. PAPER TAPE PARITY ERROR - A paper tape parity error has occurred between the PTS and PTCU. Restart program.

i. TAPE IS FILE PROTECTED - Put ring in tape and type a space.

j. WRITE PARITY ERROR - Information type-out. Write parity occurred on the tape drive. This routine will backspace and attempt to write again automatically.

k. BUS PARITY ERROR ON TAPE UNIT - Information type-out. Automatically retries.

l. TAPE UNIT NOT READY - Make tape unit ready and type a space.

m. CONNECT TCU - Tape unit typed in is not connected. Connect unit and type a space.

n. CANNOT LOCATE ICC/CICC - Insure ICC/CICC is connected. Master clear and hit run.

o. CONTROL ABORT - Open-ended prestore aborted by ACE. Restart program.

p. PRINTER OUT OF PAPER - Insert paper in printer and type a space.

q. BUS PARITY ERROR ON PRINTER - Bus parity has occurred. Notify FOCCPAC Maintenance Group.

r. CANNOT LOCATE PRINTER - Attach printer and type a space.

s. PRINTER NOT READY - Make printer ready and type a space.

t. CARD READY HOPPER EMPTY - If more cards are to be pre-stored, add the cards and type a space. To close prestore hit the carriage return.

u. CARD READ STOP - Correct condition and type a space.

v. VALIDITY ERROR - Re-position cards and type a space.

w. CARD READ ERROR - Re-position cards and type a space.

x. BUS PARITY CARD READER - Restart and inform FOCCPAC Maintenance Group.

y. CARD READER NOT READY - Make card reader ready and type a space.

z. CARD READER NOT FOUND - Attach card reader and type a space.

aa. PRESTORE ID ILLEGAL -- On card prestore, check BOF card for legal ID.

HALT WITH JUMP KEY 4 SET

A=0000   Enter tape drive or card reader select code or first two OPCON characters of PRESTORE ID. Hit run. If PRESTORE ID was entered, the second halt is to enter the second two OPCON characters of PRESTORE ID. See OPCON list for valid OPCON codes.

A=1600   Could not find paper tape stations alerted and ready. Make a station alert and ready. Hit run.

A=0020   Bus parity on PTS. Restart program.

A=0025   Reader not in five level mode. Correct and hit run.

A=0021   PT reader not ready. Make PT reader ready and alert and hit run. If no more input, clear "A" and hit run.

A=0022   Paper tape parity error. Restart program.

A=2000   Could not find an ICC unit. Connect and hit run.

A=0042   Bus parity on tape unit. Restart program.

A=0043   File protected tape. Insert ring and hit run.

A=0045   Write parity. Clear "A" to ignore. Hit run with "A" not equal $\emptyset$ to retry.

A=0055   Control aborted open-ended prestore. Restart program.

A=0073   Printer out of paper. Insert paper and hit run.

A=0072   Bus parity error on printer. Notify FOCCPAC Maintenance Group.

A=1360   Cannot locate a printer. Attach printer and hit run.

A=0071	Printer not ready. Make printer ready and hit run.
A=0063	Card read stop. Correct condition and hit run.
A=0100	Card validity error. Re-position cards and hit run.
A=0200	Card read error. Re-position cards and hit run.
A=0064	Bus parity error on the card reader. Inform FOCCPAC Maintenance Group.
A=0061	Card reader not ready. Make ready and hit run.
A=2500	Could not locate a card reader. Attach card reader and hit run.
A=0065	PRESTORE ID illegal, check BOF card for legal ID.

VALID OPCON CODES:

A=06	0=60	5=65
C=10	1=61	6=66
G=14	2=62	7=67
M=22	3=63	8=70
X=35	4=64	9=71

BOF AND EOF CARD FOR OPCON PRESTORE

1-8	15	16	17-20	
....	D	M	XMØØ	PRESTORE ID
....	D	E	XMØØ	

BOF AND EOF FOR AUTODIN PRESTORE

1-8	15	16	17-20	
....	T	M	XMØØ	PRESTORE ID
....	T	E	XMØØ	

2.5 MAGNETIC TAPE TO CARD (FILE SEGMENT 0006)

2.5.1 Function

This routine punches from 1 to 96 cards per magnetic tape record in either binary or Hollerith mode.

2.5.2 Equipment

Magnetic tape drive and card punch.

2.5.3 Options

   a. Punched cards can be either binary or Hollerith.
   b. From 1 to 9 specified files can be punched.
   c. From 1 to 9 files can be skipped before punching begins.

2.5.4   Jump Keys/Stop Keys

    a. Jump Key 4 - Set to bypass the typewriter.

2.5.5   Operating Instructions

    a. Load segment 0006 into bank 1.
    b. Master clear and hit run.
    c. Typewriter will output "MODE?" Type TC for tape to card.
    d. Typewriter will output "TAPE WXYZ". Enter WXYZ of tape
        drive.
    e. Typewriter will output "CARD WXYZ". Enter WXYZ of card
        punch.
    f. Typewriter will output "BINARY OR HOLLERITH TYPE H OR
        B". Type B for Binary or H for Hollerith.
    g. Typewriter will output "TYPE NUMBER OF FILES TO BE
        PUNCHED". Enter number of files to be punched. If all
        files are to be punched, type just the period; if a
        specified number are to be punched, enter 1-9.
    h. Typewriter will output "TYPE NUMBER OF FILES TO BE
        SKIPPED". Enter number from 0-9 to be skipped.

2.5.6   Messages/Errors

    a. MODE? - Type TC for tape to cards.

    b. TAPE WXYZ - Type WXYZ of tape drive.

    c. CARD WXYZ - Type WXYZ of card punch.

    d. BINARY OR HOLLERITH TYPE H OR B - Type H for Hollerith
        and B for Binary.

    e. TYPE NUMBER OF FILES TO BE PUNCHED - Type the number of
        files to be punched (1-9) or just a period for all files.

    f. TYPE NUMBER OF FILES TO BE SKIPPED - Type the number of
        files to be skipped, any number from 1-9.

    g. TAPE DT=.,..., ERR=.... - History information only - Tape
        data transfers and errors.

    h. CARD DT=....., ERR=.... - History information only - Card
        data transfers and errors. Errors are rejected into
        pocket 2.

    i. PROCESSING FILE NUM - Indicates that the next file is
        being processed.

    j. ROUTINE COMPLETED - PAU.

    k. CANNOT GET TAPE STATUS - Ready a tape drive.

l. HOPPER EMPTY - Add more cards to punch.

m. CANNOT GET PUNCH STATUS - Ready a card punch.

n. UNRECOVERABLE TAPE ERROR OCCURRED - Job aborted.

o. PUNCH STOP - Correct condition.

p. HIT PERIOD TO RESTART - Hit period to continue after correctable error condition such as hopper empty, punch stop, etc.

q. GIVE ME AN ICC UNIT TO ACE - Attach an ICC unit.

r. END OF TAPE SKIP ABORTED - Self explanatory.

2.6 CARDS TO PRINTER 80/80 LIST (FILE SEGMENT 0007)

2.6.1 Function

This routine will print cards 80/80 just as they are read from the reader. It also gives page numbering and total card count at the end of the listing. No classification can be obtained except from what is in the cards.

2.6.2 Equipment

Card reader and printer.

2.6.3 Options: None.

2.6.4 Jump Keys/Stop Keys

Jump Key 2 - Set for double spacing.

2.6.5 Operating Instructions

a. Load segment 0007.
b. Master clear.
c. Hit run.

2.6.6 Messages/Errors

a. P=0151 Printer not on-line. Ready printer and hit run.
b. P=0221 Reader not on-line. Ready reader and hit run.
c. P=0421 Final halt - if more runs are required, load cards and hit run.

NOTE: In the event of a card jam, remove cards from the reader, place the last card in the stacker in front of deck, and restart or run P=0243 to bypass the top of page routine. Card count includes any blank cards.

2.7.1   Function

This routine will format and output a specific area of core on to either a printer or magnetic tape.

The output will be dumped sixteen words per line in matrix form.

```
B*        00       01       02     ....      17
0000     AAAA     BBBB     CCCC             NNNN
0020     YYYY
```

AAAA = Contents of location 0000
YYYY = Contents of location 0020

If one or more lines contain the same constant, a single line will be printed giving the address range and constant value.

```
         00       01       02       03     ....      17
0120    7101     0100     0000     0000             0000
0140    thru     7776     have 0000
```

* represents bank that was dumped.

2.7.2   Equipment

A printer or tape drive on internal channel.

2.7.3   Options: None.

2.7.4   Jump Keys/Stop Keys: None.

2.7.5   Operating Instructions

a. Load segment 0011 in any bank other than the bank you wish to dump.
b. Master clear.
c. Enter parameters

    P=0110   A=000Y   Y=Bank to be dumped
    P=0111   A=BBBB   B=First word address of core to be dumped
    P=0112   A=EEEE   E=Last word address + 1 to be dumped

d. Master clear.
e. Set A=WXYO   WXYO=Select code of TCU or printer
f. Hit run.

2.7.6   Messages/Errors: None.

2.8   PAPER TAPE TO MYLAR TAPE COPIER (FILE SEGMENT 0013)

2.8.1   Function

To copy paper tape to mylar tape in the event the Tally is down. This
program has delays built in to keep the punch from over heating.

2.8.2   Equipment

Console paper tape reader and console paper tape punch.

2.8.3   Options

It is possible to make the program run faster by increasing the number
of 10-frame blocks that it reads and punches between delays (location
75 bank 0). It is not advisable to do this when punching mylar tape,
because mylar is not oil impregnated and the cycle of the punch may
cause it to heat up excessively and crack the punch pins. If it is
desired to speed up the program, the procedure is as follows:

           a.   After loading routine 13, master set P=0075.
           b.   Set "Z" = the octal number of 10-frame blocks to be
                punched between delays.
           c.   Place enter switch up.
           d.   Step computer.
           e.   Reset the enter switch.
           f.   Master clear and hit run.

2.8.4   Jump Keys/Stop Keys

Stop Key 1 - Set to stop the program.

2.8.5   Operating Instruction

           a.   Load segment 0013.
           b.   Ready console paper tape reader and console paper tape
                punch.
           c.   Master clear.
           d.   Hit run.

2.8.6   Messages/Errors: None.

2.9   DUMP DISK ON PRINTER (FILE SEGMENT 0015)

2.9.1   Function

This routine will output any track or number of tracks from disk to
a line printer.

2.9.2   Equipment

Disk file to be dumped on the external channel and a line printer on the
internal channel.

2.9.3 Options

Three print options are available:

> Option 0 = OPCON 80 character per line list
> Option 1 = 160A dump format list
> Option 2 = 1604A dump format list

2.9.4 Jump Keys/Stop Keys

Stop Key 1 - Halts to enter record length for option 0 print. P=0365 and A=0050. Set A = length of record in 160A octal words (1-74).

2.9.5 Operating Instructions

   a. Load segment 0015 into bank 0.
   b. Master clear.
   c. Hit run.
       Programmed halt P=0110   A=7000 set A = WXYM of disk file to be dumped.
   d. Hit run.
       Programmed halt P=0116   A=0001 set A = first two (2) BCD numbers of the track to be dumped.
   e. Hit run.
       Programmed halt P=0121   A=0002 set A = second two (2) BCD numbers of the first track to be dumped.
   f. Hit run.
       Programmed halt P=0145   A=5555 set A = number of tracks to be dumped in octal.
   g. Hit run.
       Programmed halt P=0151   A=1400 set A = WXYZ of line printer.
   h. Hit run.
       Programmed halt P=0156   A=0000 set A = option of printed output.
   i. Hit run.

2.9.6 Messages/Errors

   a. P-0735   Tried to read a track greater than 9999.
   b. P=1051   Lost data 64 times in one pass. Master clear and restart job.
   c. P=1076   Invalid mod or parameter given.
   d. P=0110   A=7000 Good halt - To dump more tracks do steps c through i.

The following are the BCD codes used when assigning track addressed:

Number	Enter
0	12
1	01
2	02
3	03
4	04
5	05
6	06
7	07
8	10
9	11

2.10   TELMAC PRESTORE (FILE SEGMENT 0017)

2.10.1   Function

This routine will prestore a TELMAC generated magnetic tape to an open-ended file.

2.10.2   Equipment

Magnetic tape drive, typewriter, and line printer.

2.10.3   Options: None

2.10.4   Jump Keys/Stop Keys: None.

2.10.5   Operating Instructions

a. Load segment 0017 into bank 0.
b. Master clear and hit run.
c. Typewriter will output "PRESTORE ID=". Type in PRESTORE ID of an open-ended file. If an error is made while typing, type a / (slash) to delete then retype correctly. To terminate typing, type a period.

2.10.6   Messages/Errors

a. PRESTORE ID= - Any legal PRESTORE ID of an open-ended file.

b. TAPE WXYZ - WXYZ of the tape drive containing TELMAC magnetic tape.

c. LP WXYZ - WXYZ of a line printer for on-line listing.

d. PROGRAM COMPLETE - End of program. The magnetic tape
    will rewind and unload, and program will recycle. If
    another input tape is required, hit run.

NOTE: You must wait until the auto-job is complete. This program will
automatically enable the auto-job.

e. CAN'T GET TAPE STATUS - Attach tape device and retype
    when requested.

f. CAN'T GET LP STATUS - Attach a line printer and retype
    when requested.

g. ILLEGAL - Previous type-in contained illegal information.
    Repeat type-in when requested.

h. CANNOT LOCATE ICC/CICC - Attach ICC to computer and
    hit run.

2.11   DFM DIRECTORY REBUILD (FILE SEGMENT 0035)

2.11.1   Function

a. To initialize a directory.
b. To read all directories and print them.
c. To read an individual directory.
d. To write an individual directory.

NOTE: This routine is a segment of DFM and can function only on a DFM
computer.

2.11.2   Equipment

DFC and printer.

2.11.3   Options

Described in paragraph 2.11.5 below.

2.11.4   Jump Keys/Stop Keys

Described in paragraph 2.11.5 below.

2.11.5   Operating Instructions

a. To initialize a directory.

    (1) Set P=7600 relative bank = 3

        (2)   Set all jump keys.

        (3)   Hit run.

        (4)   Halt - P=7100 A=0000
              Set A=DFC/module (module in BCD)

        (5)   Hit run.

        (6)   Good halt P=7100 A=0000

        (7)   If more directories are to be initialized,
              simply enter the DFC/module in "A" and hit run
              (go to step d).

   b.   To read all directories and print them.

        (1)   Set P=0100 relative bank = 3

        (2)   Hit run.

        (3)   Good halt P=0100 bank = 3 (DFC/module is printed in
              upper right corner)

   c.   To read an individual directory.

        (1)   Set P=7100 set relative bank = 3

        (2)   Set A=DFC/module (module in BCD)

        (3)   Hit run.

        (4)   Good halt P=7100 A=5500 (FWA of directory)

   d.   To write an individual directory.

        (1)   Set P=7100 set relative bank = 3

        (2)   Set all jump keys.

        (3)   Set A=DFC/module (module in BCD)

        (4)   Hit run.

        (5)   Good halt P=7100 A=0000

2.11.6   Messages/Errors

   a.   P=0110 A=1300   Enter printer in "A" and hit run.

   b.   A=0535   Disk read/write error - status word
                        0 in BER and status word <u>1</u> in BXR.

NOTE: Directory is written from 5500-6763 bank 3.

      Insure that a printer is on-line and ready.

2.12   PCC (PERMANENT CONTROL CARDS) (FILE SEGMENT 0036)

2.12.1   Function

To create and maintain a file of permanent control cards on the ACE
disk system. Provisions have been made to allow up to 700 decks of
control cards to reside on the disk. Segment 0036 has, by selectable
options, all services that are necessary and useful to maintain the
master file of control cards. These options are described in paragraph
2.12.3 below.

2.12.2 Equipment

DFC on external channel, card reader and printer on either channel.

2.12.3 Options

a. 0001 - Add a PCC file to the disk or replace an existing file with the same task ID. One copy of the new deck will be printed as well as a copy of the replaced deck where applicable. In case of INXX errors the file will not be placed on disk unless the override (Jump Key 1) switch is set.

b. 0002 - Itemize all existing PCC files giving (1) file ID, (2) date created/updated, and (3) an actual track address of the file. The total number of available PCC units is also printed.

c. 0003 - Print a designated file of control cards. The actual file name must be supplied and the computer will halt twice for the insertion of these parameters. For PCC files, the OPCON coded file name must be entered; for table jobs, the actual equivalent of the job number must be supplied.

   Example: PRINT PCC FILE CSP4

   ```
   1st HALT   A=0000   SET A=1030
   2nd HALT   A=0000   SET A=2564
   ```

   Example: PRINT TABLE JOB 149

   ```
   1st HALT   A=0000   SET A=0225   (149_{10}=225_8)
   2nd HALT   A=0000   SET A=0000
   ```

d. 0004 - Print a designated file of control cards including any chained data. The same parameters delineated for option 0003 must be entered.

e. 0005 - Print X copies of the INXX error messages. These messages are used in conjunction with the I/O 160-A and are provided for convenience. The computer will halt once, asking for the number of copies in the A-register.

f. 0006 - Create the PCC directory tracks from scratch. This option is only to be used when initializing a new disk and should be used only by FOCCPAC Executive Systems Group.

g. 0007 - Print the overflow directory; provide a tool for maintenance of the system and has no direct bearing on the permanent control cards.

h. 0010 - Delete a PCC task from a file. This option should be used judiciously and requires the same parameters as for option 0003.

i. 0011 - Print all PCC files.

j. 0012 - Create the overflow directory and clear mod 2 to the bottom of the table job PCC files.

2.12.4 Jump Keys/Stop Keys

Jump Key 1 - Set to override INXX errors.

2.12.5 Operating Instructions

a. Load segment 0036.
b. Master clear.
c. Hit run.
   Programmed halt   P=0120   A=0000 Enter the option desired
                                        in the A-register.

d. Hit run.

2.12.6 Messages/Errors

For all error halts reference side-by-side listing located in the Executive Systems Office or on the computer floor.

OPCON code for alpha and numeric values for operator's use when entering task name:

A=06	H=15	O=24	V=33	2=62
B=07	I=16	P=25	W=34	3=63
C=10	J=17	Q=26	X=35	4=64
D=11	K=20	R=27	Y=36	5=65
E=12	L=21	S=30	Z=37	6=66
F=13	M=22	T=31	0=60	7=67
G=14	N=23	U=32	1=61	8=70
				9=71

2.13   MNEMONIC DUMP (FILE SEGMENT 0037)

2.13.1   Function

To dump core in mnemonic form on a line printer. This routine cannot distinguish between actual data and instructions. Actual data will be interpreted as instructions.

2.13.2   Equipment

Line printer and tape drive on the internal channel.

2.13.3   Options: None.

2.13.4   Jump Keys/Stop Keys

Jump Key 2 - Set to bypass the control computer. Use when you have
no access to the control computer.

2.13.5   Operating Instructions

    a. Load segment 0037 in a bank other than the bank or
        banks to be dumped.
    b. Master clear.
    c. Hit run.
        Programmed halt   A=4000   Set A = WXYZ of tape drive
                                    where scratch is mounted.
    d. Hit run.
        Programmed halt   A=0001   Set A = Bank to be dumped.
    e. Hit run.
        Programmed halt   A=0002   Set A = First word address
                                    of area to be dumped.
    f. Hit run.
        Programmed halt   A=0003   Set A = Last word address +
                                    1 of area to be dumped.
    g. Hit run.

2.13.6   Messages/Errors

    a. A=0000   A printer was not found ready on internal channel.
    b. A=4000   Good End of Job - do steps 2.13.5c through g if
                  another area of core is to be dumped.

2.14   2 BANK PRINTITALL (FILE SEGMENT 0042)

2.14.1   Function

This segment is a magnetic tape print routine using any one of the op-
tions described in paragraph 2.14.3 below. The program can operate on
either the 8090 or 160-A.

The options that do not use format control from tape are printed in
the following standard format:

        Print labels and file marks on first page.
        Skip to top of next page.
        Print top security line.
        Advance paper one line.
        Print 56 lines of data.

Advance paper one line.
Print the page number.
Print the bottom security line.
Skip to the top of the next page.
Print the top security line, etc.
When an end-of-file label is found.
Skip to the page number line, and
Print: "PAGE XXXX OF XXXX PAGES",
Advance paper one line.
Print the bottom security line.
Skip to the top of the next page.
Print labels and file marks.
If the second label was a beginning-of-file label, reset
page counter to zero, skip to the top of the next page and
continue the print sequence.

On the last page of each file that is printed, the total number of
pages that have been printed and the page number for that page are
printed on the same line.

EXAMPLE: Page 0137 of 0137 pages.

When a file mark is encountered in the middle of a data file, the
program handles it in the following manner:

Skip to the page number line.
Print the page number.
Advance paper one line.
Print the bottom security line.
Skip to the top of the next page.
Print the top security line; and on the same line, but the
other side of the page, print: *** FILE MARK ***.
Advance paper one line and continue normal print sequence.

If format control is in use:

Skip to the top of the next page.
Print: *** FILE MARK ***.
Skip to the top of the next page and continue print operation.

2.14.2 Equipment

Tape drive and printer on internal channel.

2.14.3 Options

The following is a list of current print options provided by the
PRINTITALL utility program.

Option	Type of Print

00   Straight list - This option is a straight list, printing the first 120 characters of each record read from tape. Anything over 120 characters is ignored. If the record is less than 120 characters, but more than one (1) 160-A word long, the remaining print line will be padded with blanks. If page numbering is not deleted, this option will follow the standard print format for unformatted tapes.

01   FILEPRINT - This option is the same as option 00, except that it prints formatted tapes and checks the first position of each character. The user is responsible for the security classification.

02   OTPRINT and PFC - This option is the same as option 00, except that the control characters on tape are formatted in a different sequence.

03   Print over 120 characters - This option will print any record from 2 to 7700 octal 160-A words. This is equal to 1000 decimal 1604-A words. This option unblocks the input record one word at a time, until it has enough words to print one line, or reaches the end of the input record. It then prints a 120 character line utilizing the standard print format for unformatted tapes. After printing that line, it returns and checks to see if the entire record has been printed. If so, it reads the next input record. If there is more of the record to be printed, it continues unblocking the record one word at a time until it has printed the entire record. By unblocking one word at a time, the record may end in the middle of a printer line and still be printed correctly.

04   This option is the same as option 03 - except the line printer is advanced one line before printing each new input record. This will give the user a clear indication of where one record stopped and the next begins.

05   Binary tape dump. 160-A format binary dump. It will dump records that range in length from 2 to 7700 octal 160-A words. The paper is advanced one line between each new record, and a header is printed across the page to aid the user in reading the output. If the record takes more

Option	Type of Print

than one page to print, another header is printed
at the top of the new page. The program prints
20 octal 160-A words on each line. If the record
contains an odd number of words, not a multiple
of 20, the last line of the record will contain
from 1 to 17 octal words. Any record that is
over 7700 octal words in length will have only
the first 7700 locations printed; the rest will
be ignored. At the start of each new line of
print this option will go to a routine that
checks for repeated data.

06   Binary tape dump, 1604-A format - This option is a
1604-A format (XX X XXXXX XX X XXXXX) dump. It
will dump records that range in length from 1 to
1000 decimal 1604-A words long. The paper is
advanced one line between each new input record.
This option prints four 1604-A words per line
in instruction format. Before the line is
printed, a transfer is made to a repeated data
check routine (see option 05). If the input record
contains more than 1000 decimal 1604-A words, the
first 1000 will be printed and the rest ignored.
This option utilizes the standard print format
for unformatted tapes.

07   Straight list on 501 printer - This option is like
option 00, except it will print 136 characters
per line instead of 120 characters.

10   Over 136 characters on the 501 printer - This
option is the same as option 03, except it will
print 136 characters per line instead of 120 charac-
ters.

11   This option is the same as option 10, except it
advances one line between each new input record.

12   VARIPRNT for the OPCON printer - This option is the
same as option 03, except it checks the first
character. This option prints using the standard
print format for formatted tapes.

13   VARIPRNT for the 501 printer - This option is the
same as option 12, except it will print 136 charac-
ters per line instead of 120 characters.

14   FILEPRNT for the 501 printer - This option is the
same as option 01, except it will print 136
characters per line instead of 120 characters.

Option	Type of Print
15	FORTRAN print - Option 15 is a FORTRAN format print. It will print the first 120 characters of each record that is read from tape, with anything over that being ignored. If the record is less than 120 characters, but more than one 160-A word long, the remaining print positions of that line will be padded with blanks. If the record is one 160-A word long, it will be ignored. This option is designed to terminate the print when it locates two double period (..) records in a row.
16	DUMPEDIT for 1604-A - This option is a 1604-A format (XX X XXXXX XX X XXXXX) octal core dump. It will print four 1604-A words per line of print in instruction format, showing the first word address of this block of data and the contents of the six 1604-A index registers. Before each line is printed, this option checks for repeated data via a subroutine called equality test (see option 05). This option utilizes the standard print format for unformatted tapes.
17	Same as 16, but prints the post mortem dump (SEG ID of 00) and bypasses any restart dump.
20	Same as option 03, except it will print only 80 characters per line. This will give the user the capability to print blocked card images.

2.14.4   PRINTITALL Program Notes

Note 1: Parity Errors: The program will attempt to read a record 10 times before it considers it to have a solid parity error. The program will then do one of two things depending upon the type of print in progress.

Formatted Prints: For tapes that have control characters the program will insert **PARITY** in the last ten (10) characters of the first and/or only line of the input record.

Unformatted Tapes: For unformatted tapes, the following message will be printed:
****** TAPE PARITY ERROR ****** OCCURRED ON THE FOLLOWING RECORD ******

Note 2: Multifiled Tapes: When printing multifiled tapes, the program prints all files to the end of the reel.

2.14.5   <u>Jump Keys/Stop Keys</u>

     a.   Stop 1 - Manually enter print option. Computer will halt with P=3370 or P=0454. Clear A-register and enter desired option. Reset STOP Key 1 and hit run.

     b.   Jump Key 1 - Skip files. Computer will halt with P=5367 A=0000. Enter A with the number of files to be skipped. Reset Jump Key 1 and hit run.

     c.   Jump Key 2 - Reassign equipment. Complete any printing and then set Jump Key 2 - computer will halt with P=5215. Make new crosspoints, reset Jump Key 2 and hit run.

     d.   Jump Key 4 - Aborts a printout.

2.14.6   <u>Operating Instructions</u>

     a.   Load segment 0042 into bank 1 only.
     b.   Master clear.
     c.   Hit run.

2.14.7   <u>Messages/Errors</u>

     a.   P=3161 Mount next reel of multireel file.
     b.   P=5270 Tape drive not available.
     c.   P=5274 Printer not available.
     d.   P=5300 ICC not available.
     e.   P=0440 A=0011 No classification.

        Clear "A" and enter classification:

               0 = UNCLASS
               1 = CONFID
               2 = SECRET
               3 = TOP SECRET
               4 = TOP SECRET NOFORN
               5 = TOP SECRET SIOP ESI

     f.   P=0446 A=0012 No option on the tape - clear "A" and enter print option.

2.15   <u>MASTER DISK REBUILDER (FILE SEGMENT 0044)</u>

2.15.1   <u>Function</u>

This routine is designed to write a magnetic tape on disk for the purpose of rebuilding the system disk. Disk can also be dumped; however, central control function 27 is now utilized.

2.15.2   <u>Equipment</u>

A tape drive on internal channel with ECO switch on and a DFC on the external channel.

2.15.3   Options

a. 0000 = Dump disk to tape.
b. 0001 = Dump tape to disk.

2.15.4   Jump Keys/Stop Keys: None.

2.15.5   Operating Instructions

a. Load segment 0044 into bank 3.
b. Master clear.
c. Set relative bank to 2.
d. Hit run.
    Programmed halt   P=0103   A=0000
    Set A = option
e. Hit run.
    Programmed halt   P=0107   A=7000
    Set A = DFC and mod (e.g., 7501)
f. Hit run.
    Programmed halt   P=0125   A=4000
    Set A = tape drive being used.
g. Hit run.

2.15.6   Messages/Errors

a. A=0204   Tape error - to accept data clear "A" and hit run.
b. P=0336   A halt at the end of each reel that was dumped
             using CCC function 27. If another reel is
             required, do steps b through g of the operating
             instructions.

2.16   160-A TAPE LABELER (FILE SEGMENT 0054)

2.16.1   Function

This routine allows the operator to label magnetic tapes with the
beginning tape label.

2.16.2   Equipment

A typewriter and tape drives.

2.16.3   Options: None.

2.16.4   Jump Keys/Stop Keys

Jump Key 4 - Set to bypass the typewriter in the event that the type-
writer is down. Tape drive and reel numbers will be entered in the
A-register instead of via the typewriter.

2-24

2.16.5   Operating Instructions

a. Load segment 0054.
b. Master clear.
c. Hit run.
d. Typewriter will type out the message "WXY =" for the
   tape drive. Type the tape drive number (e.g., 611)
   followed by a period. If a typing error is made, type
   / (slash) and then the correct tape drive number
   followed by a period. After the period is typed,
   the message "REEL =" will type out. Type in the
   reel number of the reel that you wish to label
   followed by a period. Use the same procedure as above
   if a typing error is made.
e. If the typewriter is down, set Jump Key 4 and the
   computer will halt with

> A=4000   Enter A with WXYZ of the tape drive and hit
>   run.
> A=0001   Enter A with the 1st digit of reel number in
>   octal and hit run.
> A=0002   Enter A with the 2nd digit of reel number in
>   octal and hit run.
> A=0003   Enter A with the 3rd digit of reel number in
>   octal and hit run.
> A=0004   Enter A with the 4th digit of reel number in
>   octal and hit run.
> A=0005   Enter A with the 5th digit of reel number in
>   octal and hit run.

2.16.6   Messages/Errors

a. WXY=   Request for tape drive.
b. REEL=   Request for reel number.
c. Illegal   Illegal parameter entered.
d. Ring is out   Ring is out of the tape. Insert ring.
e. Not ready   Tape drive not ready.
f. Parity   Write parity.
g. Not connected Tape drive not connected.

2.17   BCD TAPE PRINT (FILE SEGMENT 0056)

2.17.1   Function

This routine allows printing of BCD tapes (even parity) at FOCCPAC.
The ECO switch may have to be turned ON in the TCU.

2.17.2   Equipment

Tape drive and printer on internal channel.

2.17.3   Options: None.

2.17.4   Jump Keys/Stop Keys: None.

2.17.5   Operating Instructions

        a. Load segment 0056.
        b. Master clear.
        c. Set the P-register to 110 and enter the tape drive
            select code in the Z-register.
        d. Step the computer.
        e. Enter the printer select code in the Z-register.
        f. Step the computer.
        g. Master clear and hit run.

2.17.6   Messages/Errors

Check the side-by-side listing for halts.

2.18   160-A SSEC BOOTSTRAP (FILE SEGMENT 0066)

2.18.1   Function

This is not literally a nonsystem routine as such. It serves the purpose
of getting one more bootstrap off the computer floor and onto disk
where it is more readily accessible. It is used to call the program
off the SSEC tape.

2.18.2   Equipment

Magnetic tape drive on either channel.

2.18.3   Options: None.

2.18.4   Jump Keys/Stop Keys

Jump Key 1 - Set to cause the computer to halt to allow the operator
to enter the magnetic tape drive select code and channel where the
SSEC master tape is hanging.

2.18.5   Operating Instructions

        a. Load segment 0066.
        b. Master clear.
        c. Set Jump Key 1.
        d. Hit run.
            Programmed halt A=0006   Set A=WXYO of tape drive where
                                      the SSEC master is.
        e. Hit run.
            Programmed halt   Enter the A-register with the
                                      channel of the tape drive where
                                      the SSEC master is on.

f. Hit run.
    Programmed halt A=0000   Enter the A-register with the
                              select code of the reader.
g. Take down Jump Key 1.
h. Hit run to start the SSEC-controlled task.

2.18.6   Messages/Errors

None. If a halt occurs after card deck has started to read, refer to
NAVCOSSACT Document No. 12M901D TR-01, Appendix Z (MCSIM and SSEC halts).

2.19   NSU BOOTSTRAP (FILE SEGMENT 0067)

2.19.1   Function

Load enough instructions to call segment 0000 into core.

2.19.2   Equipment

Console paper tape reader.

2.19.3   Options: None.

2.19.4   Jump Keys/Stop Keys: None.

2.19.5   Operating Instructions

a. Load bootstrap (MYLAR) tape in desired bank.
b. Good halt.
    P=0370   A=4040
    If any difference, repeat steps a and b.

2.19.6   Messages/Errors: None.

2.20   8090 MONITOR (FILE SEGMENT 0074)

2.20.1   Function

To provide an interface between the 8090 and ACE control using the
161 typewriter. Also required with segment 0076 to get the grasp
specs from disk.

2.20.2   Equipment

161 typewriter.

2.20.3   Options: None.

2.20.4   Jump Keys/Stop Keys: None.

2.20.5 Operating Instructions

    a. Load segment 0074 into bank 1.
    b. Master clear.
    c. Hit run.
    d. Typewriter will output "8090 MONITOR", hit carriage return.
    e. Typewriter will output "CON FILE ID", type in desired segment number followed by a period.

    NOTE: If an error was made while typing in the segment number, retype the correct segment number after typing a / (slash).

2.20.6 Messages/Errors

    a. "8090 MONITOR" indicates that segment 0074 is loaded.
    b. "CON FILE ID" Type in desired segment number.

2.21 GRASP SPECIFICATION COMPILER (FILE SEGMENT 0075)

2.21.1 Function

To compile OCR page reading specification on tape in the format expected by segment 0076 (Grasp prestore to disk program).

2.21.2 Equipment

Tape drive, 161 typewriter and 915 page reader.

2.21.3 Options: None.

2.21.4 Jump Keys/Stop Keys: None.

2.21.5 Operating Instructions

    a. Load segment 0074.
    b. After segment 0074 types out "CON FILE ID" type in "75".
    c. Typewriter will output "REEL NO". Enter the reel number where the specs are to be compiled. (Tape must be mounted on drive 0.)

    NOTE: If tape output of the compiled specification is not desired, hit space and go to step f.

    d. Typewriter will output "FILE ID". Type in file identification followed by a period. File ID must be 15 characters or less. (Steps c and d physically relabel the tape.)
    e. Typewriter will output "SEG ID". Type disk segment identification of which the last three digits must match the last three digits of the second parameter in the "DOC" control field of the specification page.

If compilation is complete, hit space and control will return to the 8090 monitor.

f. Typewriter will output "PRESTORE ID". Type in open-ended file ID to which the user's job will be prestored.

g. Program will be in an idle loop waiting to read the specification page. A load-specification "bottom of the page" will terminate reading of that page. An "E" will terminate the reading of a set of specification and return you to step e.

2.21.6 Messages/Errors

a. Reel No   Type reel number
b. File ID   Type file ID, example "SPEX"
c. Seg ID   Type disk segment ID
d. Prestore ID   Type actual open-ended file ID where segment 0076 will prestore to.

Type a space to close the output tape.

2.22   GRASP PRESTORE (FILE SEGMENT 0076)

2.22.1   Function

To read OCR pages, pull the specification from disk and prestore the text to an open-ended file. The prestore ID on the top of the OCR pages indicates to segment 0076 which specs to use and which open-ended file to prestore to.

2.22.2   Equipment

915 page reader and 161 typewriter.

2.22.3   Options: None.

2.22.4   Jump Keys/Stop Keys: None.

2.22.5   Operating Instructions

a. Load segment 0074 into bank 1.
b. Master clear and hit run.
c. Typewriter will output "8090 MONITOR", hit carriage return.
d. Typewriter will output "CON FILE ID", type in "76".
e. Program will be in an idle loop waiting for the pages to be prestored.
f. As each page is prestored, the word "FORM" and the four character form number is typed out.

2.22.6   Messages/Errors

a.   **Form XXXX**   Indicates the open-ended file to which
                     the page will be prestored.

b.   **Pres ID Err**   Page reader couldn't read the form line.
                     This is usually caused by no space
                     between the line locate bar and prestore
                     ID or page didn't load correctly at
                     initial read point.

NOTE: Unreadable character will result in the typing of
       the entire line with an equal = sign replacing the
       bad characters. If you can read the rejected charac-
       ters, type them in and hit the carriage return.

Example: AB=DEF - the = sign is "C". Type in the "C" and
          hit carriage return.

NOTE: If you have more than one = sign on any line, you
       must type in all the characters after the first =
       sign, then hit the carriage return.

Example: AB=D=FG - the = signs are "C" and "E". Type
          CDEFG and then hit carriage return.

NOTE: If you have an = sign and there are two characters
       missing, you must type from the = sign.

Example: AB=EFG - the = sign is "CD" which should be two
          characters. Type CDEFG and hit the carriage
          return.

NOTE: If you cannot read a rejected character, hit backspace.
       This will cause the entire line to be rejected and
       the next line to be read.

SECTION III

1604A SYSTEM UTILITIES

3-1.1

SECTION III

1604A SYSTEM UTILITIES

3.1   **IPS FILE FORMAT TO JOVIAL TABLE DEFINITION (JOT)**

3.1.1   **Program Description**

3.1.1.1   Language: JOVIAL

3.1.1.2   Function

Converts data definitions found in an IPS Master File Format record to
a format acceptable to the JOVIAL compiler or the Compool assembly
program.

3.1.1.3   Option

JOT may be operated under control of the Program Production System
(PPS) with control information taken from control cards. JOT may also
be operated by SSEC with a standard set of parameters assumed by the
program.

3.1.1.4   Entry to Program

Preloaded by SSEC or loaded by PPS along with control cards. See
paragraph 3.1.2 for control information format.

3.1.1.5   Requirements/Limitations

No checks are performed to determine if the input tape is of the
proper format. If it is not, the results will be incorrect. Control
cards are not error checked.

3.1.1.6   Linkage

        a.   Input

            Input to JOT is the File Format Table and the Item Title
Table records from an existing IPS-format master file.
These tables are FFFC and FSTC as described in NAVCOSSACT
Document No. 12M901D TR-02, Information Processing
System (IPS) User's Manual. Additionally, optional
JOT control information may be provided when JOT is
operated by the PPS. See paragraph 3.1.2 for control
information format.

b. Output

Output from JOT is a file that contains data definitions in PPS format and the necessary control information to update a symbolic file with these definitions.

c. Error Stops/Messages

JOT uses the system abnormal termination procedure for errors detected in I/O operations.

3.1.1.7   Printouts: None.

3.1.1.8   Method

Information systems at FOCCPAC use data bases that are maintained and queried with the Information Processing System (IPS). Some reports, due to their format and frequency of operations, can be more efficiently produced by the use of single purpose report generators rather than by the use of IPS. One of the tasks required in producing a report generator is that of defining file data in the language of the report generator. Program JOT has been written to perform this task for the programmer by converting data definitions found in an IPS Master File Format record to a format acceptable to the JOVIAL compiler or the Compool assembly program.

JOT reads from file 05 two records which are assumed to be the File Format record and the Item Title record of an IPS master file. A PPS table definition is written onto file 28. This definition includes an item definition of each item and subitem defined in the File Format Table. Set names, repeat level reference numbers, and item titles are written as comments. Imbedded blanks in item names are changed to single primes since imbedded blanks are illegal in PPS item names. When the Compool format is requested for output, item names are truncated to the first eight characters. Reel number and file name from the input tape are also printed as comments on the output tape. PPS control cards are written on the output tape so that it may be used to update a user's program with cards containing data definitions from the File Format Table.

3.1.1.9   Subroutines Required: N/A.

3.1.1.10 Tables and Files: See paragraph 3.1.1.6.

3.1.1.11 Peripheral Equipment: Magnetic tape used for input and output.

3.1.1.12 Storage Locations: 15000 through 27150.

3.1.1.13 Sample Task Deck

a. Time, Job and Task cards are to be prepared in accordance with NAVCOSSACT Document No. 12M901D TR-01, Appendix H.

b. File Cards

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	30	File code of this utility program.
5-6	MT	Device type (mag tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U	Classification of this file.
15-16	MM	Indicates that this file is multi-filed to SSEC Master.
20	P	Preload indicator.
27-40	PPSIII	File identification of PPS Master.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	29	File code of card file.
5-6	CR	Device type (card reader) assigned to this file.
7	I	Use (input) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXX	File identification of card file. (user's ID)

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	28	File code of this file.
5-6	MT	Device type (mag tape) assigned to this file.
7	0	Use (output) of this file.

COLUMNS	CONTENTS	EXPLANATION
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of PPS table definition (user's ID).
62-63	XX	Number of save days.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	27	File code of this file.
5-6	MT	Device type (mag tape) assigned.
7	S	Use (scratch) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of the user's history file.
49-50	XX	Print option desired.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	25	File code of this file.
5-6	MT	Device (mag tape) assigned.
7	O	Use (output) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of user's binary program file.
62-63	XX	Save days (14 minimum).

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	21	File code of this file.
5-6	MT	Device (mag tape) assigned.

7	O	Use (output) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of user's new symbolic file.

Note: File 17 (disc) and File 18 (typewriter) are required in accordance with existing compile instructions.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	11	File code of this file.
5-6	MT	Device (mag tape) assigned.
7	P	Use (protected input) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of user's old symbolic file.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	07	File code of this file.
5-6	MT	Device (mag tape) assigned.
7	P	Use (protected input) of this file.
9	U, C, S, T O, E or I	Classification of this file.
15-16	MM	This file is multifiled to SSEC Master disc.
27-40	JOT	File identification of JOT binary program.
53-57	18069	Reel number of this tape.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as file card.
3-4	05	File code of this file.
5-6	MT	Device (mag tape) assigned.
7	P	Use (protected input) of this file.
9	U, C, S, T O, E or I	Classification of this file.
27-40	XXXXXXXXXXXXXX	File identification of user's FMS input file (containing the file format record and the item title record).

c. Data cards of card file

1) All data cards shown in paragraph 3.1.3 should be included.
2) Data card XGOTO is explained in paragraph 3.1.2.
3) Other cards are required as part of standard PPS requirements.

3.1.2   JOT Control Information

Control information is entered by the XGOTO function of PPS, a function which sets the PPS control table to the values on the XGOTO card and causes a jump to the specified address. The control card is in the following format:

XGOTO JOT tablename tablelength n_1 cardnumber n_2 type 15000 $

XGOTO	PPS function
JOT	Identifies this card as JOT input
tablename	8 OPCON characters used as the name of the user's table output by JOT.
tablelength	An integer of 4 characters or less, specifying "number of entries."
n_1	An increment added to the starting word location of item descriptions output by JOT.
cardnumber	A 3 letter, 5 digit identifier used to sequence output.

n_2 A 1 to 5 digit integer used to increment card number.

type   When this parameter is coded as CLASS, JOT will write output in COMPOOL format which includes table and item security classification. Item names will be truncated if greater than 8 characters.

COMPOOL FORMAT

```
SYSUP FROM-11 TO-21  $
ALTERS BEGIN
INSERT$ RH(RG009100)    $
TABLE OPST02  V 35n S 1 S SSMASTER RECORD TABLE      27 JAN 1967    SSRG009200
REGIN$$OPST02 $$   SS TAPE NO. 43796  FILE INTERIMBUFF902           SSRG009300
ITEM NSETS      48 U 01 0  U SS NUMBER-OF-ENTRIES                   SSRG009400
ITEM REPLEV      6 U 02 0  U SS REPEAT LEVEL                        SSRG009500
ITEM FIXCOUNT   14 U 02 6  U SS NO. FIXED LENGTH ITEMS              SSRG009600
ITEM VARCOUNT   14 U 02 20 U SS TOTAL NO. OF ITEMS                  SSRG009700
ITEM SETLOC     14 U 02 34 U SS RELATIVE LOCATION OF SET            SSRG009800
SS********** SET 0 FIXED    REPEAT LEVEL 0 **************           SSRG009900
ITEM UNIT   H 12 001 0 U S SSUNIT       SS                          RG010000
ITEM UNIT1  H 02 001 0 S S SSUNIT1      SS                          RG010100

SS********** SET 01 ARMAMENT REPEAT LEVEL 01 ****************       SSRG023200
ITEM MEOPT  H 11 001 0 S S SSMEOPT      SS                          RG023300
ITEM DIN    H 01 001 0 S S SSDIN        SS                          RG023400

SS********** SET 01 PLANS REPEAT LEVEL 02 ****************          SSRG026300
ITEM PLAN   H 06 001 0 S S SSPLAN       SS                          RG026400
ITEM PLAN1  H 01 001 0 S S SSPLAN1      SS                          RG026500

SS********** SET 01 SIOP REPEAT LEVEL 03 ****************           SSRG028200
ITEM SORTI  H 25 001 0 S S SSSORTI STATUS  SS                       RG028300
ITEM SDNUM  H 04 001 0 S S SSNUM        SS                          RG028400

SS********** SET 01 TREATY REPEAT LEVEL 04 ****************         SSRG028900
ITEM TRTY   H 06 001 0 S S SSTREATY     SS                          RG029000
SS********** SET 01 CASREPT REPEAT LEVEL 05 ****************        SSRG029100
ITEM FOCODE    H 05 001 0 S S SSFCODE   SS                          RG029200
ITEM EQUIPSDE  H 25 001 30 S S SSEQUIP  SS                          RG029300

END SSOPST02 SS
ALTERS END                                                         RG030500
CARD $
```

JOVIAL FORMAT

```
SYMUP IROM=11 TO=21  $
ALTERS BEGIN
INSERT= RH(RG009100)  $
TABLE OPST01  V 350 S 1  $$MASTER RECORD TABLE    27-JAN-1967    $$RG009200
         BEGIN$$UPST01  $$   $$ TAPE NO. 43796   FILE  INTERIMBUFF902  $$RG009300
ITEM NSETS      I 48 U 01   0   $   $$ NUMBER-OF-ENTRIES        $$RG009400
ITEM REPLEV     I  6 U 02   0   A   $$ REPEAT LEVEL             $$RG009500
ITEM FIXCOUNT   I 14 U 02   6   $   $$ NO, FIXED LENGTH ITEMS   $$RG009600
ITEM VARCOUNT   I 14 U 02  20   $   $$ TOTAL NO. OF ITEMS       $$RG009700
ITEM SETLOC     I 14 U 02  34   $   $$ RELATIVE LOCATION OF SET $$RG009800
$$**********   SET  0  FIXED   REPEAT LEVEL  0  ********** $$RG009900
ITEM UNIT       H 12  001   0   $   $$UNIT    $$   RG010000
ITEM UNIT1      H 02  001   0   $   $$UNIT1   $$   RG010100

$$**********   SET 01 ARMAMENT  REPEAT LEVEL 01  ********** $$RG023200
ITEM MEOPT      H 11  001   0   $   $$MEOPT   $$   RG023300
ITEM DIN        H 01  001   0   $   $$DIN     $$   RG02 400

$$**********   SET 01 PLANS     REPEAT LEVEL 02  ********** $$RG026300
ITEM PLAN       H 06  001   0   $   $$PLAN    $$   RG026400
ITEM PLAN1      H 01  001   0   $   $$PLAN1   $$   RG026500

$$**********   SET 01 SIOP      REPEAT LEVEL 03  ********** $$RG028200
ITEM SORTI      H 25  001   0   $   $$SORTI STATUS  $$   RG028300
ITEM SONUM      H 04  001   0   $   $$NUM     $$   RG028400

$$**********   SET 01 TREATY    REPEAT LEVEL 04  ********** $$RG028900
ITEM TRTY       H 06  001   0   $   $$TREATY  $$   RG029000
$$**********   SET 01 CASHEPT   REPEAT LEVEL 05  ********** $$RG029100
ITEM EUCODE     H 05  001   0   $   $$ECODE   $$   RG029200
ITEM EQUIPSDESC H 25  001  30   $   $$EQUIP   $$   RG029300

     END $$OPST01  $$
ALTERS END                                        RG030500
CARD  $
```

3.2   NAVASSEMBLER (1604A)

3.2.1   Language: CODAP

3.2.2   Function

The function of NAVASSEMBLER is to assemble any number of prestored
CODAP coded programs that are non-Compool sensitive into loader control
binary format.

3.2.3   Options

    a. If a library is to be used, it must be declared as file
        number 14.
    b. If no library is needed, only four files must be declared;
        viz., 01, 11, 12 and 13.

3.2.4   Entry to Program

The program is nonrelocatable and originates at location 12000 octal.
It is preloaded by the System and control is transferred to it by
leaving the starting address blank on the Task card.

3.2.5   Linkage

    a. Input

        File 1:   NAVASSEMBLER - preloaded.
        File 11:   Card image tape.
        File 14:   Library tape if desired.

    b. Output

        File 12:   Side-by-side tape.
        File 13:   Binary tape.

    c. Error Termination Codes

        After an abnormal termination, an error message will
        appear in the form 37XX fc YY addr PPPPP; when XX
        represents the status response for the file in question,
        YY represents the File Code and PPPPP represents the
        octal location of the I/O calling sequence.

        In addition, TERM REAS 3735 indicates that an exces-
        sive number of EQU's have been processed.

3.2.6   Requirements/Limitations: None.

3.2.7   Peripheral Equipment

Magnetic tape drives for input files and magnetic tape drives for output files are required.

3.2.8   Storage Locations

The program is originated at 12000 octal. Approximately 10500 (octal) locations are utilized.

3.2.9   Sample Task Deck

a. Time, Job and Task cards are to be prepared in accordance with NAVCOSSACT Document No. 12M901D TR-01, Appendix H.

b. File Cards

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	01	File code of this utility program.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U	Security classification of this file.
15-16	MM	Indicates that this file is multifiled to the SSEC Master.
20	P	Preload Indicator
27-40	ASM	File identification name of utility program.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	11	File code of card image tape.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.

| 9 | U, C, S, T, O, E or I | Security classification of this file. |

| 27-40 | XXXXXXXXXXXXXX | File identification name of card image tape. |

| 53-57 | XXXXX | Leave blank when running at FOCCPAC. |

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	12	File code for side-by-side tape.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	O	Use (output) of this file.
9	U, C, S, T, O, E or I	Security classification of this file.
27-40	XXXXXXXXXXXXXX	File identification name for side-by-side tape.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	13	File code for binary tape.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	O	Use (output) of this file.
9	U, C, S, T O, E or 1	Security classification of this file.
27-40	XXXXXXXXXXXXXX	File identification name for Binary tape.

COLUMNS	CONTENTS	EXPLANATION
*1	F	Identifies card as a file card.
3-4	14	File code of library tape.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.

* This file is to be used only if a library is needed.

9	U		Security classification of this file.
15-16	MM		Indicates that this file is multifiled to the SSEC Master.
27-40	LIBRARY		File identification name for the Library tape.

3.2.10   Prestored Deck Structure

CARD	CARD TYPE	COLUMNS	CONTENTS
1st	IDENT	10-14	IDENT
		20-23	XXXX Program ID
Last for this Seg	TRANSFER	10-12	END
		20-28	Label of starting address

Repeat the above-mentioned cards for each program to be assembled.

Last for this group of programs (optional)	FINIS	10-14	FINIS signals NAVASSEMBLER that all programs have been assembled. End-of-file produces same effect.

3.2.11   Pseudo Operation Codes Available

The NAVASSEMBLER and PPS Assembly Program have basically the same capabilities. The Pseudo Operation Codes listed below are all available with the NAVASSEMBLER. A detailed description of the codes may be found in the PPS Assembly Language section of the Program Production System User's Manual.

1. BES	9. FDC	17. ORGR
2. BESZ	10. FINIS	18. REM
3. BSS	11. FLX	19. RST
4. BSSZ	12. HED	20. SPACES
5. DEC	13. IDENT	21. TEL
6. EJECT	14. LIB	22. TYP
7. END	15. OCT	23. WST
8. EQU	16. ORG	

3.3   SNOCTALF - (1604A OCTAL AND FDC (OPCON CODE) CARD LOADER
        AND SNAPSHOT CORE DUMP ROUTINE)

3.3.1   Language: CODAP

3.3.2   Function

SNOCTALF may be incorporated into any 1604A program to provide the
capability of utilizing an unlimited number of Octal Correction, FDC
(OPCON code) and/or up to 20 distinct Snapshot Core Dump Cards per
computer run. The routine expands some of the debugging capabilities
provided by the MCS/MCSIM and PPS Systems. The features of SNOCTALF
which make it differ from either or both of the System debugging
packages are as follows:

   a. SNOCTALF will accept an unlimited number of Octal
       Correction Cards for any given run. These cards
       are in the same format as the MCS/MCSIM Run Modification
       Cards. Each card can contain octal corrections for up
       to four consecutive memory locations. SNOCTALF will
       also accept an unlimited number of cards which contain
       FDC (OPCON code) data. Each card can contain OPCON
       data for up to nine consecutive memory locations.
       Both types of cards may be input in the same job and
       they may be in any order.

   b. The routine requires only about 500 (octal) memory
       locations. The equipment needed consists of a card
       file and one tape file for the snapshot dumps.

   c. When using SNOCTALF, no separate debugging monitor is
       introduced. The routine functions with the object
       program and operates completely within the MCS/MCSIM
       environment. No special types of control cards are
       required to gain access to SNOCTALF.

   d. With SNOCTALF, dynamic dumps (snapshots) may be inserted
       at execution time in any desired program locations.
       The snapshots need not be planted in the symbolic deck
       and assembled with the object program.

   e. SNOCTALF also provides the capability for limiting the
       number of successive passes through the snapshot
       location which will produce the dump, and for permitting
       preliminary passes through this location before dumping
       begins. If no limit is specified and the object program
       should inadvertently loop through a snapshot location,
       SNOCTALF will automatically terminate dumping after
       63 passes.

f. With SNOCTALF, any single Snapshot Core Dump Card may
    define up to four distinct areas of core to dump each
    time the snapshot is executed.

g. Index registers, the AC and MQ, are preserved both at
    the initial card processing entrance and at the
    occurrence of each snap.

3.3.3   Options

SNOCTALF is available to the user as a self-contained, relocatable
program file which may be preloaded into any area of core not used
by the MCS/MCSIM System or the object program. The routine is located
on the same tape or disk as the other 1604A System Utility programs.
Program options are as described in paragraph 3.3.2 above.

3.3.4   Entry to Program

SNOCTALF is designated as a preloaded file from the 1604A System
Utility Master. Both SNOCTALF and the object program are preloaded
concurrently into core. Control is given first to SNOCTALF which
places the desired octal corrections, FDC (OPCON Code) data and/or
snapshots in the object program. Control then goes to the octal
location in the object program which is punched in columns 76-80 of
the SNOCTALF Termination Card and normal processing is begun. No
RTJ SNCTLF calling sequence is required in the object program when
SNOCTALF is used in this manner. When using SNOCTALF, some modifi-
cations are necessary in the MCS/MCSIM control card deck of the object
program. (See paragraph 3.3.5g.)

3.3.5   Linkage

SNOCTALF will accept Octal Correction, FDC (OPCON Code) and/or Snap-
shot Core Dump Cards as input. These cards are placed in the card
reader at run time and are read when SNOCTALF is executed. The three
card types may occur in any order within the input card deck.

Slight modifications to the MCS/MCSIM control deck for the object
program are necessary when using SNOCTALF. These changes are also
described in this section.

a. Octal Correction Card Format

    (Same as MCS/MCSIM Run Modification Card.)

COLUMNS	CONTENTS	EXPLANATION
1	5	Identifies card as Octal Correc-tion (Run Modification) Card.
2-6	Octal address of first word location.	

7-22	Octal contents of first word. *	
23-38	Octal contents of second word *	*If any field is left blank, successive fields are not interpreted.
39-54	Octal contents of third word. *	
55-70	Octal contents of fourth word.*	
71-80		Ignored

b. FDC (OPCON Code) Card Format

COLUMNS	CONTENTS	EXPLANATION
1	4	Identifies card as FDC Modification Card.
2-6	Octal Address of first word location.	
7-8	May be 01 through 09	Number of words of FDC data to be inserted.
9-16	FDC data for first word.	
17-24	FDC data for second word.	
.........etc.........		
73-80	FDC data for ninth word.	

c. Snapshot Core Dump Card Format

COLUMNS	CONTENTS	EXPLANATION
1	6	Identifies card as Snapshot Core Dump Card.
2-6	(Octal)	Location where snap is to be implanted.

7-8	(Octal)	Number of passes to be made through this location before dumping begins. (Blanks=00.) Use leading zero.
9-10	(Octal)	Maximum number of consecutive passes through this location for which dumping is desired. (Blanks=77.) Use leading zero.
11-15	(Octal)	First word address (FWA) of core area to be dumped.
16-20	(Octal)	Last word address +1 (LWA+1) of core area to be dumped.
21-30	(Octal)	2nd set of dump parameters for this location. (Blank if not needed.)
31-40	(Octal)	3rd set.
41-50	(Octal)	4th set.
53-56		Format control characters for each of the four dump areas corresponding to format control character (F) in word a+2 of MCS/MCSIM Dump calling sequence. A blank is equivalent to 0 (Octal Format). At present, only octal format will be produced. As additional capabilities are added to the MCS/MCSIM dump, they will automatically become available with SNOCTALF.

SNOCTALF uses the MCS/MCSIM Dump routine; the actual dump will be taken before execution of the instruction at the specified snap location. SNOCTALF will provide an ID Code for each dump. This code will consist of the two OPCON characters formed by the last four octal digits of the snap location address.

d. SNOCTALF Termination Card

This card is placed at the end of the Octal Correction and/or Snapshot Core Dump Cards to indicate to SNOCTALF that no more cards are to be processed. Columns 76-80 have a special function as noted below.

COLUMNS	CONTENTS	EXPLANATION
1-75	(Blank)	These columns must be left blank.
76-80	(Octal)	These columns contain the starting address within the object program to which SNOCTALF will transfer at completion of its card processing.

e. Beginning-File Label Card

The SNOCTALF input deck (Octal Correction, FDC (OPCON Code) and/or Snapshot Core Dump Cards) must be preceded by this standard MCS/MCSIM control card.

COLUMNS	FIELD
1-8	Periods
16	M
17-30	User's Choice
31-40	Date
49-56	Task Name

f. Ending-File Label Card

The SNOCTALF Termination Card must be followed by this standard MCS/MCSIM control card.

COLUMNS	FIELD
1-8	Periods
16	E
17-30	User's Choice

g. MCS/MCSIM Control Deck Modifications

The following modifications must be made to the object program MCS/MCSIM control card deck to make use of SNOCTALF.

(1) Task Card

The Task Card must contain as its Starting Address the address into which SNOCTALF is to be preloaded (columns 19-23).

This is the same as the address in columns 21-25
of the SNOCTALF program file card.

If snapshots are being taken, a "D" is required in
column 13 of the Task Card to indicate dumping.
Any post-mortem dumps specified on the Task Card
in columns 31-41 will appear on the same dump
tape with the snapshots. This tape may be printed
on the 160A using the 160A System Utility program
DUMPEDIT.

(2) Card Reader File Card

SNOCTALF always requires a card reader input file.
Therefore, a File Card for a card reader file must
be added to all MCS/MCSIM control card decks.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies Card as File Card.
3-4	10	File Code Number
5-6	CR	Devise type (Card Reader) assigned to this file.
7	1	Use (Input) of this file.
9	U, C, S, T, O, E or I	Security Classification of this file.
27-40		Card Reader File ID

(3) SNOCTALF Program File Card

A File Card for the SNOCTALF program is required.
Its format is as follows:

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies Card as File Card.
3-4		File Code (To be chosen by the user).
5-6	MT	Devise type (Magnetic Tape) assigned to this file.
7	P	Use (Protected Input) of this file.

9	U	Security Classification (unclassified) of this file.
15-16	MM	Indicates that this file is multifiled to the SSEC Master.
20	P	Preload Indicator.
21-25	(Octal)	SNOCTALF Relocation Address. Address, not used by MCS/MCSIM or object program, into which SNOCTALF is to be loaded. This is the same address which appears in columns 19-23 of the Task Card.
27-40	SNCT	File Identification Name.

(4) Dump File Card

A dump file must be declared whenever snapshots are being taken by SNOCTALF. If only octal corrections and/or FDC (OPCON Code) cards are being used, no dump file has to be declared. The format of the dump file card is as follows:

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies Card as a File Card.
3-4		File Code. (To be chosen by the user.)
5-6	MT	Devise type (Magnetic Tape) assigned to this file.
7	O	Use (Output) of this file.
9	U, C, S, T, O, E or I	Security classification of this file. U - Unclassified C - Confidential S - Secret T - Top Secret O - SIOP E - Sensitive I - Supersensitive
18	D	Specifies this File as a Dump File.
27-40		Dump File identification name.

3.3.6   Printouts

There are no operational printouts. Term Reason No. 3730 indicates that an error associated with the card reader has occurred. In this case, the task is aborted.

3.3.7   Requirements/Limitations

    a. No limit is placed on the number of Octal Correction and/or FDC (OPCON Code) Cards. A maximum of 20 Snapshot Core Dump Cards will be processed.

    b. Caution should be exercised in the choice of snap locations. SNOCTALF will replace the complete word at that address with an RTJ and a NOP.

    c. No error indication is given in the current version for a non-octal character on an input card. The character will be converted Mod 8.

    d. Since SNOCTALF uses the MCS/MCSIM Dump routine, space should be made available for it when calculating the lowest origin for the object program.

3.3.8   Subroutines Required: MCS/MCSIM Dump Service. (SSEC)

3.3.9   Peripheral Equipment

One tape drive (only when snapshot dumps are being taken), card reader.

3.3.10   Storage Locations: Approximately 500 (octal).

3.4   1604A TRACE PROGRAM (SYSTRACE)

3.4.1   Program Description

3.4.1.1   Language: OPCON CODAP for 1604A Phase 2

3.4.1.2   Function

The Trace Program provides services for the user to locate and diagnose program errors and a means of correcting them with a minimum of programming effort. To accomplish this function, Trace Program makes available to the user the following four distinct services:

    a. Modify

        Modifying program segments of a task as they are dynamically loaded and/or overlaid for their execution.

b. Trace

Tracing and recording the execution flow of a task
or part of it.

c. Monitor

Monitoring and recording the step by step execution
of a task or part of it.

d. Snap

Taking selective dumps of core storage areas and machine
registers at specified points and times during the
execution of a task.

All these services provided by the Trace Program may be easily utilized
without any recompilation and/or reassembly of any program segment of
the task.

The Trace Program is fully compatible with the Master Control and 1604A
Subsystem Executive Control of the OPCON Operating System. All file
control, input-output, and loader services are accomplished through
MCS/MCSIM and SSEC.

3.4.1.3 Options

a. The modification of any program of a task prior to its
    execution may be requested by octal correction cards
    identical in format to the MCS/MCSIM run modification
    card.

b. Any desired program segment areas, or complete range
    of any program segment, or all program segments which
    are loaded and or overlaid dynamically during the execu-
    tion of a task may be traced and/or dumped.

c. A count-conditional provides the user with a means of
    qualifying the time when a trace should be activated
    or when a dump is to be taken.

d. The trace or dump requests for relocatable program
    segments may be specified as relocatable addresses.

e. The trace of all executed instructions (Monitor), or
    the trace of only those instructions which change
    execution sequence and path of a program segment
    (Trace) may be requested.

f. SSEC calling sequences within a program segment area
    which is being traced will not cause the trace to be
    terminated. When these calling sequences are executed,

the trace will be temporarily suspended until the return from SSEC, and then will be continued.

g. A dump (Snap) request may specify the dump of two desired core storage areas. The dump list for each request includes the dump of machine registers and core locations 0-17 inclusive.

h. The user may specify the amount of maximum output to be produced by the Trace Program. When this maximum output is produced, the execution of the task may be continued without any more significance of the trace and dump requests, or the task may be terminated (see page 3-28).

3.4.1.4 Entry to Program

The Trace Program is initiated by a standard control cards deck under either MCS or MCSIM. The Trace Program segment is preloaded through the preload indicator of the file card for SYSTRACE, and after initialization by SSEC, control is relinquished to the Trace Program at the starting address as given on the task card. The task starting address must be identical to the preload address as given on SYSTRACE File Card. The following is a general structure of a control cards deck for a task utilizing Trace Program services for an object program(s) set. (See NAVCOSSACT Report No. 12M901D TR-01, Appendix H, Control Card Formats.)

1) Time Card (MCSIM only)

2) Job Card

3) Task Card (starting address of Trace Program in columns 19-23)

4) File Card (Trace Program File - preload)
   File Card (Trace Requests File - see below)
   File Card (Trace Program output file)
   File Card(s) (Object program file(s) - preload at
                least one)
   File Card(s) (Object program input file(s))
   File Card(s) (Object program output file(s))
   (Optional D - cards must follow their respective
   F - cards. There must be one F - card for each file
   and one C - card for each change in configuration
   required.)

5) Optional Run Modification Cards (maximum of 8)

6) Sentinel Card (blank card if in MCSIM mode)

3.4.1.5   Linkage

a.   Input

In addition to the control cards deck required by
the Master Control or bv the Master Control Simulator
a Trace Requests File is required as input by the
Trace Program.

Trace Requests File (Control Cards Input File)

All requests for correction, trace and dump for a given
task are grouped together into a Trace Requests File.
This file may be on tape in card format. The file
code number assigned to the Trace Request File is 25.
If the Trace Request File is to be entered through a
file code other than 25, the Trace Program must be
modified by a MCS/MCSIM modification card for this new
code. This modification card must be as shown below:

COLUMNS	CONTENTS	EXPLANATION
1	5	Identifies card as modification card.
2-6	Octal SYSTRACE Preloading address as given on Task and SYSTRACE File cards plus 1	
7-20	Zeros	
21-22	Octal File Code for the Trace Request File	
23-70	Blanks	
71-80		Ignored.

A Trace Requests File is headed by the control card
*TRACE. A minimum Trace Requests File may consist of
only *TRACE control card. Such a minimum file initiates
complete monitoring or tracing of all programs dynami-
cally loaded and/or overlaid during the execution of
a task. Paragraph 3.4.2 describes in detail these and
all other card types which may be included in a Trace
Requests File. A generalized Trace Requests File is
shown in paragraph 3.4.3.

b. Output

The formats of output from Trace Program for Trace, Monitor, and Snap are shown in paragraph 3.4.4. All output writing is buffered and overlapped with the execution of Trace Program and/or object program(s) execution (see page 3-28 for maximum output).

3.4.1.6 Printouts

No operation printouts. If any error occurs, a core dump is taken and error code is given in the dump. (See paragraph 3.4.5 for Error Codes.)

3.4.1.7 Method

The Trace Program after receiving control from SSEC, initializes the program controls, and then transfers control to the initializer.

The initializer reads in the Trace Requests File and converts, translates, packs and then stores contents of the control cards in the allocated areas. Specifically, the initializer constructs the following three tables:

1) Load and Overlay Control Table

2) Correction Table

3) Trace, Monitor and Snap Table

After constructing the tables, the initializer relinquishes control to the executor.

The executor controls all object program(s), and immediately after they are dynamically loaded and overlayed, it corrects them according to the correction table and then it inserts Trace/Monitor/Snap requests as calls to itself and then gives control to the object program. If and when these inserted calls are executed, Trace/Monitor is activated and/or dumps are taken according to the Trace and Monitor table, and or Snap table. This interaction and transfer of control between Trace Program and the object program(s) continues until the task is terminated by the object program itself or when permissible maximum or specified output is produced by SYSTRACE.

3.4.1.8 Requirements/Limitations

a. A Trace requests file may not contain more than a maximum number of trace or dump requests (see paragraph 3.4.6).

b. Trace/Monitor areas may neither be nested nor overlapped.

c. Only one Snap may be requested at any given program point.

d. Trace/Monitor or Snap may not be requested at the following program locations:

(1) Locations which are modified dynamically during the execution.

(2) Locations which are referred by the return jump instructions.

(3) Locations which are within the calling sequences.

(4) Locations which contain the return jump instructions.

e. Any execution within a Trace/Monitor area which does not pass through the specified beginning address will neither activate Trace/Monitor nor increase the count for count-conditional. If such execution passes through the specified end address, a remark in regard to this execution will be written on the Trace Program output file.

f. Any return jump or calling sequence other than SSEC calling sequences within a program segment area which is being traced will cause the trace to be terminated if the jump address is not within the specified trace area.

3.4.1.9   Subroutines Required: None.

3.4.1.10   Tables and Files

The Trace Program is contained in a program file called SYSTRACE. SYSTRACE may be on a multifile reel.

3.4.1.11   Peripheral Equipment

1 Magnetic tape or card reader (for Trace Requests File).

1 Magnetic tape (for SYSTRACE Output File).

1 Magnetic tape (for SYSTRACE Program File. It may be on a multifile reel with other system or object program files.)

Other peripheral equipment as required by object program(s) and as indicated by file cards.

3.4.1.12   Storage Locations

SYSTRACE is relocatable and requires 5606(octal)/2950(decimal) core locations.

All preloaded or dynamically overlaid program segments of the task must be loaded above core location 11475(octal)/4925(decimal).

3.4.2   Description of Trace Request File Control Cards

3.4.2.1   *TRACE Card
(Fields that are explicitly required are underlined.)

COLUMNS	CONTENTS	EXPLANATION
1-6	*TRACE	Identifies card as the primary control card of the Trace Requests File.
7-9		Not interpreted.
10-13	Segment Identifier IDEN	Segment identifier of the pre-loaded program segment to which initial control must be given by the Trace Program for the object program(s) execution (left justified).
14		Not interpreted.
15-19	First Address FIRST	Absolute octal address of the first location of the preloaded program segment shown in columns 10-13 (5 octal digits).
20		Not interpreted.
21-25	Last Address LASTA	Absolute octal address of the last location of the preloaded program segment shown in columns 10-13 (5 octal digits).
26		Not interpreted.
27-31	Starting Address START	Absolute octal address to which initial control must be given by the Trace Program (5 octal digits).
32		Not interpreted.
33-34	Trace Program File Code FT	File code for SYSTRACE Program segment (right justified decimal number). If it is left blank, File Code 23 will be assumed.

COLUMNS	CONTENTS	EXPLANATION
35		Not interpreted.
36-37	Trace Program Output File Code FO	File code for the Trace Program output (right justified decimal number). If it is left blank, File Code 24 will be assumed.
38		Not interpreted.
39	Trace Type M/T	Optional. Type of trace if all task to be traced. M = MONITOR (complete trace) T = TRACE (flow trace) If it is left blank, T will be assumed.
40		Not interpreted.
41-45	Maximum MXOUT	Optional. The maximum number of printed lines to be permitted as the trace and/or dump output for this task. When this maximum output is produced, the trace and/or dump requests will have no more significance. It is a decimal number and must be smaller than 12000. If it is left blank, the system value 12000 print lines will be assumed (right justified decimal number).
46		Not interpreted.
47-50	Termination TERM	Optional. When the maximum output specified through columns 41-45 is produced, the execution of the task will be terminated if columns 47-50 contain TERM. If it is left blank, the execution of the task will be continued without any trace and dump until it is terminated by the object program itself.
61		Not interpreted.
62-75		Not interpreted.
76-80	Serial No.	EAM use.

3.4.2.2   L-Card*
           (Fields that are explicitly required are underlined.)

COLUMNS	CONTENTS	EXPLANATION
1	L	Identifies card as preloaded program card.
2-9		Not interpreted.
10-13	Segment Identifier IDEN	Segment identifier of the pre-loaded program segment. There must be one L-card for each program segment which is pre-loaded (except for the program segment shown, columns 10-13 of *TRACE Control card) (left justified).
14		Not interpreted.
15-19	First Address FIRST	Absolute octal address of the first location of the preloaded program segment shown in columns 10-13 (5 octal digits).
20		Not interpreted.
21-25	Last Address LASTA	Absolute octal address of the last location of the preloaded program segment shown in columns 10-13 (5 octal digits).
26-75		Not interpreted.
76-80	Serial No.	EAM use.

*If the information to be supplied by L-cards can be
obtained from SSEC, no L-card will be required for
the preloaded program segments.

3.4.2.3   P-Card
           (Fields that are explicitly required are underlined.)

COLUMNS	CONTENTS	EXPLANATION
1	P	Identifies card as Program Identi-fication card.
2-9		Not interpreted.
10-13	Segment Identifier IDEN	Segment identifier of the program for which the services of the Trace Program must be provided according to the associated 5-cards (Octal Modification Cards) and M-cards (left justified).
14		Not interpreted.
15	Absolute/ Relocatable	A = For Non-Relocatable Program R = For Relocatable Program
16-75	A/R	Not interpreted.
76-80	Serial No.	EAM Use.

3.4.2.4   5-Card (Octal Modification Card)
           (Fields that are explicitly required are underlined.)

COLUMNS	CONTENTS	EXPLANATION
1	5	Identifies card as modification card.
2-6	Octal Address of First Word Location	
7-22	Octal Contents of First Word	
23-38	Octal Contents of Second Word	
39-54	Octal Contents of Third Word	
55-70	Octal Contents of Fourth Word	
71-80		Ignored.

(If any field is blank, successive fields are not interpreted.
All location references are non-relocatable (absolute)
addresses.)

3.4.2.5   M-Card
(Fields that are explicitly required are underlined.)

COLUMNS	CONTENTS	EXPLANATION
1	M	Identifies card as Trace/Monitor/Snap Card.
2-9		Not interpreted.
10	T/M/S	T = Trace M = Monitor S = Snap
11		Not interpreted.
12-16	Octal Address OCTAL	If column 10 = T/M: Starting point to Trace or Monitor. It is an absolute or relative address according to the preceding P-card. The content of this point must not be modified dynamically during the execution of the segment which is associated with this M-card. It may not be a location within another Trace/Monitor area, e.g., Trace or Monitor areas may neither be overlapped nor nested. If column 10 = S: Snap point Dumps are performed before execution of the instructions at this point. The content of this point must not be modified dynamically during the execution of the segment. Only one snap may be requested at any point. The snap points may be within Trace or Monitor areas (5 octal digits). However, the snap points may not be at the beginning or end point of Trace or Monitor areas.
17		Not interpreted.

COLUMNS	CONTENTS	EXPLANATION
18-22	Octal Address OCTAL	If column 10 = T/M: End of Trace/Monitor area. It may not be a location within another Trace/Monitor Area (5 octal digits). The content of this point must not be modified dynamically during the execution. If column 10 = S: Not interpreted.
23		Not interpreted.
24-28	I DECML	Optional. I for Count-conditional.
29		Not interpreted.
30-34	J DECML	Optional. J for Count-conditional.
35		Not interpreted.
36-40	K DECML	Optional. K for Count-conditional. (Trace/Monitor is activated or snap is taken the Ith time and every jth time thereafter up to and including Kth time whenever program step at the location shown in columns 12-16 is executed. Any execution within a Trace/Monitor Area which does not pass through the location shown in columns 12-16 will neither activate Trace/Monitor nor change Count-conditional. I, J, and K are right justified decimal numbers and they must be smaller than 32768. If K is blank, no termination count will be imposed for blank I or J (1) will be assumed.)

COLUMNS	CONTENTS	EXPLANATION
41-75		If column 10 = T/M:
		Not interpreted.
		If column 10 = S (Optional. If no dump area is specified, all core locations are dumped.)
41		Not interpreted.
42-46	Absolute Octal Address OCTAL	Beginning of first dump area.
47		Not interpreted.
48-52	Absolute Octal Address OCTAL	End of first dump area.
53		Not interpreted.
54-58	Absolute Octal Address OCTAL	Beginning of second dump area.
59		Not interpreted.
60-64	Absolute Octal Address OCTAL	End of second dump area
65-75		Not interpreted.
75-80	Serial No.	EAM use.

3.4.3   A Generalized Trace Requests File

(Beginning-of-File Label)

*TRACE Card

L-Card

L-Card
.
.
.
L-Card

P-Card

5-Card

5-Card
.
.
.
5-Card

M-Card

M-Card
.
.
.
M-Card

P-Card
.
.
.
P-Card
.
.
.
(End-of-File Label)

Generalized Maximum Layout of Trace Control Cards

(Fields that are explicitly required are under-
lined, *TRACE and *TEND control cards must be
the first and the last card respectively.)

*TRACE   IDEN,FIRST,LASTA,START,FT,FO,M,MXOUT,TERM

L   IDEN,FIRST,LASTA

L   IDEN,FIRST,LASTA
.
.
.
L   IDEN,FIRST,LASTA

P   IDEN(R)

5FIRSTFIRST OCTAL WORDSECND OCTAL WORDTHIRD OCTAL WORDFORTH OCTAL WORD

5FIRSTFIRST OCTAL WORDSECND OCTAL WORD
.
.
.
5FIRSTFIRST OCTAL WORD

M   T,OCTAL-OCTAL,DECML,DECML,DECML

M   M,OCTAL-OCTAL,DECML,DECML,DECML
.
.
.
M   S,OCTAL   DECML,DECML,DECML,OCTAL-OCTAL,OCTAL-OCTAL

P   IDEN(A)
.
.
.
P   IDEN(R)
.
.
.
*TEND

3.4.4   Trace Program Output Formats

3.4.4.1   Output Format for Snap (Dump) Requests

PRGA(R),SNAP,00027,COUNT=84,77100-77200,77600-77777

	A UPPER	A LOWER	Q UPPER	Q LOWER	B1	B2	B3	B4	B5	B6
00000014	00000036	7700007	7777777	01134	01140	00000	00011	00003	00000	

Addr								
00000	76 0 00000	00 0 77770	00 0 05205	00 0 05260	00 0 25763	00 0 26002		
00004	00 0 26002	00 0 25754	77 7 77573	77 7 77573	75 0 05615	75 0 00024		
77100	20 0 77541	07 0 00060	11 0 00003	41 0 77335	61 0 77424	75 0 77104	12 0 77541	76 0 77100
77104	01 0 00030	41 0 77335	61 0 77423	01 0 00025	20 0 77611	75 0 77110	75 0 00000	76 0 77107
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
77200	75 0 77141	50 0 00000	75 4 77203	50 0 00000	75 0 77152	50 0 00000	75 0 77200	12 0 77611
77600	67 6 76067	63 0 50621	21 0 51024	23 3 10616	23 0 56765	05 6 40560	60 0 06261	05 6 76505
77604	64 0 56060	60 6 26105	00 0 07777	77 7 77777	66 6 66666	66 6 66666	06 0 50505	05 0 50505

LOCATIONS 77610 THROUGH 77637 ALL CONTAIN   00 0 00000   00 0 00000   00 0 00000

LOCATIONS 77640 THROUGH 77773 ALL CONTAIN   66 6 66666   66 6 66666   66 6 66666

| 77774 | 05 0 50505 | 05 0 50505 | 05 0 50505 | 05 0 50505 | 05 0 50505 | 05 0 50505 |

3.4.4.2 Output Format for Monitor Requests

PRGB(R),MONITOR,00000-00100,COUNT=15

LOC	MNE	OP	B	M-FLD	A UPPER A LOWER	Q UPPER Q LOWER	B1	B2	B3	B4	B5	B6
15000U	LDA	12	2	35042	0000001400000035	7770007777777777	00000	12020	00000	00011	00003	00000
15000L	SAL	61	0	35073	0000001400000035	7770007777777777	00000	12020	00000	00011	00003	00000
15001U	ARS	01	0	00030	0000000000000014	7770007777777777	00000	12020	00000	00011	00003	00000
.....	...	..	.	.....	....	....	.....	.....	.....	.....	.....	.....
.....	...	..	.	.....	....	....	.....	.....	.....	.....	.....	.....
.....	...	..	.	.....	....	....	.....	.....	.....	.....	.....	.....
15100L	ENA	10	0	00036	0000000000000036	0000000000000000	12020	01140	00000	00011	00003	00000

3.4.4.3   Output Format for Trace Requests

PRGC(A),TRACE,20100-20300,COUNT=125

LOC	MNE	OP	B	M-FLD	A UPPER A LOWER	Q UPPER Q LOWER	B1	B2	B3	B4	B5	B6
20100U	RTJ	75	4	20134	00000000000000000	7777707777777777	00000	00001	00005	00146	00000	77777
20134L	SLJ	75	0	20205	00000000000000000	7777707777777777	00000	00001	00005	00146	00000	77777
20205L	AJP	22	0	20212	00000000000000000	7777707777777777	00000	00001	00005	00146	00000	77777
20212L	QJP	23	1	20310	00000000000000000	7777707777777777	00000	00001	00005	00146	00000	77777

3.4.5   Description of Error Codes

Octal Error Code	Description of Error
10	Buffer type not defined.
11	Attach type neither write nor read.
12	Read request for a non-attached file code.
13	No stacked read available, possibly read buffers not released.
14	Release routine error.
15	A stack read request for a non-attached buffer pool.
16	Check write request for a non-attached buffer.
17	Embedded check write routine finds no stacked write.
20	Last+1 of just terminated write not equal to stacked write.
21	Write request for a non-attached file code.
22	STACKR finds read buffer error.
23 not used	
24 not used	
25	Trace/monitor request for a non-identified program.
* 26	Trace/monitor range not within program boundaries.
* 27	LOC1/LOC2 lower instruction contains a conditional jump instruction.

* Illegal use of SYSTRACE.

Octal Error Code	Description of Error
* 30	Trace program output greater than permissible maximum output; execution is terminated (TESTC Routine).
31	Possible error in TESTC routine.
* 32	First control card not *TRACE card.
* 33	LAOCT overflow; number of preloaded programs greater than permissible maximum number.
34	Possible error in B10 sequence.
* 35	Trace program output greater than permissible maximum output; execution is terminated (WRITEOUT Routine).
36	Segment mark on TRACECC file.
37	Buffer error while reading from TRACECC file.
40	Tape parity or other error while reading from TRACECC file.
41	TRACEOUT file protected.
42	TRACEOUT file end-of-tape.
43	Tape parity error while writing on TRACEOUT file.
44	Other error while writing on TRACEOUT file.
* 45	Illegal control card.
* 46	Illegal character in (P) control card.
* 47	Trace, Monitor and Snap table overflow, more than permissible number of M-cards are encountered.
* 50	Illegal character in OPCON decimal field
51	Error in binary to OPCON decimal conversion, argument greater than permissible maximum.

* Illegal use of SYSTRACE.

Octal Error Code		Description of Error
* 52		Illegal character on (M) control card.
53		Unsuccessful loading of executor segment.
* 54		Correction table overflow.
55		Possible error in calling sequence to IDENSCAN.
* 56		Trace/monitor request within trace/monitor area.
57		Fault return address cannot be determined.
* 60	not used	
* 61		Column 39 of *TRACE card neither T/M nor blank.
62		Illegal code at (SNAPFLAG).
63		Illegal security classification.
64	not used	
65		Possible error in reactivating trace/monitor.
* 66		Exit from preloaded program range while tracing all task.

3.4.6   Requirements and Limitations

Card Type	Permissible maximum number of card types for each Trace Request File
L	19
C	20
M	20

* Illegal use of SYSTRACE.

3.5   TAPECONVERTBIN (1604A FOREIGN TAPE CONVERSION)

3.5.1   Program Description

3.5.1.1   Language: JOVIAL

3.5.1.2   Function

The function of TAPECONVERTBIN is to convert BCD or Binary records
from 1401, 1410, 704 or 7090 created magnetic tapes to OPCON-coded
records written as a file on an OPCON-system-labeled magnetic tape.

3.5.1.3   Options

    a. Input Tape Parity

        The input tape(s) may be ODD parity (Binary) or
        EVEN parity (BCD). EVEN parity is assumed, unless
        the user otherwise specifies. See paragraphs
        3.5.1.6b and 3.5.2.

    b. Input Tape Positioning

        The input tape can be positioned before conversion
        by specifying the number of segment/tape marks to
        be skipped. Any skipping of labels or record count
        data must be requested by the user via the parameter
        card.

    c. Multireel Input

        Up to 11 reels of magnetic tape may be converted.
        File 1 must be declared as the first reel; then files
        21-30 in consecutive ascending order. The output
        from this option will be a multireel file.

    d. Restart Capability

        This option is available and a complete core dump
        is taken after reading each 1000 input records.

3.5.1.4   Entry to Program

The program operates as a task under normal SSEC procedures and
standard system control cards are required. This program is origi-
nated relocatable at 00000; therefore, the relocation address must
be specified on file card for FILE 10.

3.5.1.5   Linkage

a. Input

File 01 - Foreign tape to be converted - H or L in
         column 26 of file card indicates tape density.

File 03 - Card Reader for Parameter Card. See para-
         graph 3.5.2.

File 10 - Program file - preload. Columns 21-25 of
         file card must contain <u>relocation address</u>,
         <u>TAPECONVERTBIN</u>.

File 21-30 - Foreign tapes to be converted in addition
            to File 01. Used only if more than one
            tape is to be converted.

b. Output

File 02 - Converted tape.

File 20 - Restart tape, if this feature is desired.

c. Error Termination Codes

All messages are routed through the SSEC abnormal
termination path.

0401 - Insufficient FILE cards or nonconsecutive file
       numbers declared for input reels (Files 21-30).
       Given when attempting to convert more than one
       tape and the required file cards have not been
       declared.

37<u>XX</u> - Where <u>XX</u> is the input/output status given back
       by SSEC when an error occurs. These statuses
       are identical to those in NAVCOSSACT Document
       No. 12M901D TR-01, Appendix M.

3.5.1.6   Sample Task Deck

a. Control Cards

(1) Time, Job and Task cards are to be prepared in
     accordance with instructions in Appendix H of
     NAVCOSSACT Document No. 12M901D TR-01, Appendix H.

(2) File Cards

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	01	File code of first tape to be converted.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U, C, S, T, O, E or I	Security classification of this file.
26	H or L	Denotes density of foreign tape.
53-57		Leave blank when running at FOCCPAC.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	02	File code of converted tape.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	0	Use (output) of this file.
9	U, C, S, T, O, E or I	Security classification of this file.
27-40		File identification name of converted tape.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	03	File code of card file.
5-6	CR	Device type (card reader) assigned to this file.
7	I	Use (input) of this file.
9	U, C, S, T, O, E or I	Security classification of this file.

NOTE: For each additional foreign tape to be converted, it is necessary to prepare a file card as above using the file code numbers 21 through 30.

| 27-40 | | File identification name of card file. |

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	10	File code of this utility program.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U	Security classification of this file.
15-16	MM	Indicates that this file is multi-filed to the SSEC Master.
20	P	Indicates that first segment of this file is to be preloaded.
21-25		Relocation constant-indicates address at which to begin to load this program.
27-40	FORN	File identification name of program file.

b. Parameter Card - This card is always required.

COLUMNS	CONTENT
1-2	Segment/tape marks to be skipped before converting. Blar = 0, otherwise 01-99. NOTE: If tape to be converted is a 1401 tape starting with a file 0, put 02 in these columns.
5-8	Number of characters per input record. 01-7680 right justified. This field is ignored if SAME does not appear in columns 13-16; program will write the same amount of characters that it reads.

NOTE: This parametric card is to be placed between a
       beginning-of-file label card and an end-of-file
       label card. See NAVCOSSACT Document No. 12M901D
       TR-01, Appendix I, paragraphs I.2.2.1 and I.2.2.2.

9-11	ODD, if Tape to be converted (File 01) was generated in Odd parity. Blank = even parity.
13-16	SAME, if all input records are the same size and the size is given in columns 5-8; otherwise, leave blank.
37-40	1401 or 1410, if File 01 is a 1401 or 1410 tape, respectively; blank, if File 01 is a 7090 or 704 tape.
79-80	02 - 11 - Total number of reels when File 01 is multireel; otherwise, leave blank.

3.5.2 Characteristics of Card Image Tapes Generated on the 1401 and 704 Tapes at CINCLANT

Tape Characteristics	704	1401
Parity	Odd	Even
Density	Low	High
No. of Characters	72	84
No. of Characters to be placed on Parameter Card	80	80
Segments to be Skipped Initially	0	0 or 2

Other Information:

a. The card to tape program used on the 704 creates an extraneous record (000200000020000..0200) in the middle of the tape. This record is converted by TAPECONVERTBIN and must be deleted subsequently by an Updater program (SYMUP, CARD/UPDATER, TPUPDATE) before the tape can be used successfully for an assembly or compilation.

b. Since the 704 writes only 72 characters and TAPECONVERTBIN will write 80, the sequence field (characters 73-80) of the converted tape (File 02) will contain garbage. The resequencing feature of SYMUP and CARD/UPDATER will easily alleviate this inconvenience.

c. There are two different card to tape programs on the
1401 at CINCLANT. One automatically puts a file 0 on
each tape and the other starts putting card images on
starting at load point. It is this author's understanding
that the latter type will become the standard. If file
0, however, is generated, the user must request the tape
to be skipped forward 2 tape marks. See paragraph
3.5.1.6b columns 1-2.

3.6 UNLIMITED OCTAL CORRECTIONS

3.6.1 Program Description

3.6.1.1 Language: 1604A absolute code

3.6.1.2 Function

To allow an unlimited number of octal corrections in the system format
to be read in from the 1402 card reader.

3.6.1.3 Options: None.

3.6.1.4 Entry to Subroutine

The program is on seven octal correction cards in the MCS/MCSIM deck.
Entry to program is gained by the transfer address on the task card;
columns 19-23 must contain 77701. After all cards are processed,
control is transferred back to the USER'S PROGRAM at the address
specified in columns 10-14 of the first card of the octal corrections
program. See paragraph 3.6.2 for the seven octal program cards.

3.6.1.5 Requirements/Limitations

a. USER'S PROGRAM may not occupy locations 77700-77745.
b. No error checks are made for nonoctal punches on USER's
OCTAL CORRECTIONS.

3.6.1.6 Linkage

a. Input: OCTAL CORRECTIONS in usual format.
b. Output: Corrections are made to core.

3.6.1.7 Printouts: None.

3.6.1.8 Method

Cards are read in one at a time and processed. When a blank card
is found, control is transferred back to the USER'S PROGRAM.

3.6.1.9 Subroutines: None.

3.6.1.10   Tables and Files

A ten-word table for the card input area is located at 77734-77745.
The card file is located in the 1402 card reader. It must contain:
Beginning-of-file and End-of-file cards, the USER'S OCTAL CORRECTIONS,
and a blank card to terminate processing.

3.6.1.11   Peripheral Equipment: 1402 card reader.

3.6.1.12   Storage Locations: 77700-77745.

3.6.1.13   Status: Tested and documented.

3.6.1.14   Classification: Unclassified.

3.6.2   Octal Program Cards

The below cards contain the program, "Unlimited Octal Corrections."

USER'S PROGRAM   OCTAL EQUIVALENT
STARTING ADDRESS*****   OF FILE CODE **

    5,77700,750*****,50000000,100000**,61077703,61077730,61077705,

    75400154,01000000,

    5,77704,75400142,00000000,00000002,00000000,00000000,00000000,

    00077734,00077746,

    5,77710,75400143,00077704,12077734,22277730,50100000,75477722,

    03000036,61077720,

    5,77714,07000006,61077727,50100001,50000000,75477722,50000000,

    16077727,61077727.

    5,77720,07000052,21000000,72077720,75077716,75000000,50200001,

    16177734,51100001,

    5,77724,23277704,50600007,06000003,07000003,55677725,55277723,

    75077722,00000000

    5,77730,75400154,02000000,75077700,50000000,

NOTE: Commas indicate individual instructions and neither commas nor
       spaces may be punched on the cards.

3.7   ECOP (1604A FOREIGN, EVEN PARITY TAPE COPY)

3.7.1   Language: JOVIAL

3.7.2   Function

The function of ECOP is to copy foreign even parity magnetic tapes
and convert any 00 codes to 77 codes to allow processing in the
OPCONCTR.

3.7.3   Options

a. Record size may be from 1 to 4125 (octal) 1604 words.
b. Input records containing permanent parity errors may be
    skipped.
c. Input tapes containing no 00 codes may bypass this code
    check resulting in a 40% decrease in copy time.

3.7.4   Entry to Program

The program operates as a task under normal SSEC procedures, and standard
system control cards are required.

3.7.5   Linkage

a. Input

    File 01 - foreign tape to be copied.

    File 03 - option card (must be included, even if no
              option is selected).

    File 04 - program file (file ID is ECOP, preload from
              master disk).

b. Output

    File 02 - OPCONCTR compatible copy.

    Typewriter - copy status.

    (1) For all successful copies the following comment
         is typed: total records NNNNNNNN largest rec
         NNNNNNNN smallest rec NNNNNNNN where:

         (a) NNNNNNNN is octal
         (b) total records = records successfully copied.
         (c) largest rec/smallest rec = record length in
              1604 words.

3-49

(2) When the skip option is selected, the following comment is typed: parity on input record NNNNNNNN where:

   (a) NNNNNNNN is octal
   (b) a total of 25 parity skips are allowed. The copy terminates if more are encountered.

c. Error Termination Codes

   (1) 37XX - where XX is the input/output status given back by SSEC when an error occurs. These statuses are identical to those in NAVCOSSACT Document No. 12M901D TR-01, Appendix M.

   (2) 7775 - SSEC write parity status (occurs when the NONE option is selected and a 00 code is encountered).

3.7.6   Task Deck

a. Control cards are to be prepared in accordance with instructions in Appendix H of NAVCOSSACT Document No. 12M901D TR-01 (FOCCPAC 1604-A and 160-A SSEC/MCSIM User's Guide).

b. Option card must be blank for standard copy or contain:

   (1) SKIP - in columns 1-4 if permanent parity records are to be skipped. (NOTE: If permanent parity is encountered and this option is not selected, the task will terminate with an SSEC 3705 reason.)

   (2) NONE - in columns 9-12 if no 00 codes will be encountered. (It is recommended that only tapes which are copies generated by this routine and that are being copied again use this option.)

c. A sample task deck using both the SKIP and NONE option appears on the following page.

```
A
J ?OU SUTILITYF17S1    LANCH 1G01
T 01TFST               101504
F 01MTP U              HINPUT/TAPE
F 02MTO U              HOUTPUT/TAPE
F 03CRI U              OPTION/CARD     14
F 04MTP U        D     FCOP/RIN/A

*******        MOPTION/CARD
SKIP   NONE
*******        EOPTION/CARD
```

3.8 BADSAP (BADSAP ASSEMBLY PROGRAM)

3.8.1 Language: JOVIAL

3.8.2 Function

BADSAP is a three-pass assembly program which converts programs written in symbolic notation (BADSAP/OSAP) to a fixed, machine language program. BADSAP is written to operate on an OPCON 1604, assembling programs for OPCON 160 execution.

3.8.3 Options

a. Symbolic input may be via cards, system tape, or foreign tape.
b. A card image tape of any segment may be generated as output.
c. An octal-format card deck of any segment may be generated as output.
d. All segments may be resequenced.
e. Symbols may be defined as common to all segments or as unique to a particular segment.

Additional information on these options may be found in the sections on LINKAGE and METHOD.

3.8.4 Entry to Program

Entry to BADSAP is made using standard system control cards.

3.8.5 Linkage: Optimum multifiling is as noted:

a. Input

    File 01 - The old master, created in a previous BADSAP generate or update run. This file must be declared only when running BADSAP in an update configuration.

    File 03 - BADSAP control and SYM cards. This file may be cards or a card image magnetic tape.

    File 13 - Library file. This file is optional; it has the format of a BADSAP master.

    File 30 - BADSAP program file. This file must be protected input, preloaded, and multifiled to the Master (P in column 7, P in column 20, MM in columns 15-16).

b. Output

File 02 - The new master, created in a generate or an update run.

File 04 - Side-by-side list.

File 05 - Magnetic tape binary output. This should be multifiled to File 02 (02 in columns 15-16). NOTE: This file cannot be preloaded. It must be loaded using either the CDC 160A utility BADSAP loader or special bootstrap.

File 06 - Octal-format card deck output of any segment. This file is optional and should be declared only if an octal deck is desired.

File 07 - Card image magnetic tape extract of any segment. This file is optional and should be declared only if a card image segment extract is desired. If declared, File 07 should be multifiled to File 01 (01 in columns 15-16).

File 10 - Serial disk (100 cylinders), used as scratch.

File 11 - Serial disk (100 cylinders), used as scratch. This should be multifiled to File 10.

File 12 - Random disk (2 cylinders), used as scratch. This should be multifiled to File 10.

File 15 - Dump file. This file should be multifiled to File 04.

c. Abnormal Termination Codes

BADSAP uses three Abnormal Termination Codes - 3700, 3705, and 3777. These termination codes are associated with I/O errors. With either error the file code associated with the Abnormal Termination will be included. The 3700 error is produced when the subsystem executive returns 20 consecutive unrecoverable write parities on a single record of the BADSAP assembly. The only files which can produce the 3700 termination are:

The New Master (File 02)
The Side-by-Side Tape (File 04)
The Binary Object Program (File 05)

The 3705 code represents an unrecoverable read parity error on File 02, the new master, during the second pass. The 3777 code represents

an unrecoverable read parity error on File 13, the library. I/O
error checking is not performed by BADSAP for the following files:

File 03 - The card or card image input to BADSAP.
File 06 - The card punch.
File 07 - The card image output tape.
File 10 - The interim symbol table.
File 11 - The interim Binary.
File 12 - COMPOOL Card Image Buffer.

The old master (File 01), in the event of either a tape parity or an
invalid NENT word, is processed in the following fashion:

The last SEG ID read correctly from the old master will be placed
in all thirty entries of the master input table and a sequence number
will be generated for the thirty entries using the sequence number
read from the previous record as a starting base. Additionally, all
entries in the table will have their OPCODE descriptions changed to
REMARKS. The CHANGE INDICATOR FIELD, which is normally blanked on
input, will be filled with a "T" for "tape error." Once this has
been accomplished, BADSAP returns to its normal update sequence. All
records flagged in this manner will be listed as errors on the output
side-by-side with the SEG ID and Sequence Number for subsequent cor-
rection in another run.

3.8.6   Printouts

a.   The BADSAP Assembly System will produce a complete
     side-by-side of all segments during a generate run.
     During an update run only those lines of a segment which
     are in error will be listed, unless that segment was
     changed during that update run, in which case a side-
     by-side will be produced for the entire segment. For
     those users familiar with the standard OSAP output, the
     side-by-side produced by BADSAP will represent a slight
     change. The binary generated by two-word instructions
     will be listed on a single line. The FDC Pseudo Op
     in BADSAP representation only requires one line on the
     side-by-side because the binary for the first word only
     is listed.

     (1) LISTALL

          The LISTALL control is used to override the auto-
          matic suppression of side-by-side production of those
          segments that were not changed. Thus, the use of
          LISTALL will cause all segments on the master to
          be listed.

(2) OLDLIST

The OLDLIST control card allows the user to obtain
the standard OSAP form of the side-by-side.

b. If the SYMTAPE control is utilized and File 07 is
declared, a card image extract of segments from the
master will be generated. As many segments as desired
may be extracted and printed.

3.8.7 Method

a. Description of Passes

(1) PASS 1 - INPUT PROCESSING

It is in PASS 1 that all of the symbolic input
and control cards with a BADSAP Assembler are pro-
cessed. The update control cards (CHANGE, INSERT,
and DELETE) have a required sequence which is
dependent upon the Master being updated. The
remaining control cards (OCTDECK, SYMTAPE,
LISTALL, FDCCODE, LIBRARY, LIBLIST, BCDCODE,
SEABASS, SAVDATE, BENCHMK, and OLDLIST) are merely
stored in PASS 1 and affect the operation during
PASS 2.

(a) Symbolic Analysis

All cards inputted to the BADSAP Assembly
System which do not contain, starting in
column 1, the previously listed control words
will be assumed to be OSAP Coding. BADSAP,
unlike the OSAP Assembler, requires that all
LABELS, OPCODES, and M-TERMS start in the
first position of the appropriate field. As
the symbolic card is analyzed, the M-TERM is
unpacked using (+) and (-) signs as separators.
A BLANK ( ) is used as the scan termination
character in BADSAP. BADSAP will accept up
to four (4) symbols in the M-TERM. During the
analysis of the symbolic input, the instruc-
tion type and the number of locations required
are determined. Symbolic analysis is per-
formed only on initial input.

(b) Symbolic Update

In normal operation the BADSAP Assembly System
uses as input an old master which contains in
compiler-allocated, variable length, multiple-
entry tables the results of all symbolic

3-55

analysis performed on preceding runs. The
master also contains in each record a segment
identifier and sequence number. These two
fields are used in conjunction with the
CHANGE, DELETE, and INSERT control cards for
updating the master.

(c) Symbol Table Generation

As the new master is generated in PASS 1,
calculations are performed to assign the label
fields their equivalent core position. BADSAP
will record each label, its core location,
and the defining card number in a symbol table.

1 Segment Symbol Table (SST)

Each segment processed by BADSAP has its
own SEGMENT SYMBOL TABLE (SST); uniqueness
of labels between segments thus can be
guaranteed. The SST is stored intermedi-
ately on serial disk for usage in PASS 2,
and is listed following its segment on the
side-by-side.

2 Common Symbol Table (CST)

BADSAP has an option in which a COMMON
SYMBOL TABLE (CST) may be generated for
usage in all segments. A symbol is
inserted into the CST by placing a "C"
in column 8 of the card which defines the
symbol. Symbols defined in this manner
may be referenced in any segment. The
CST is listed separately at the end of the
side-by-side output; it is labeled
"COMPOOL LISTING". The information provided
in the CST list includes:

a The identity of the segment which
   defined the label.

b The card sequence number.

c An indication of multiple definitions
   of symbols.

d An indication of reference made to a
   symbol.

e The index to the card image held on
   random disk (File 12).

3-56

The CST is held in core throughout the run.

(d) Segment Set/Use Table

<u>1</u> Generation

The Set/Use Table is also generated in PASS
1 for each segment <u>processed</u> by BADSAP.
The Set/Use Table will appear on the
side-by-side listing following the Segment
Symbol Table listing. This table will be
labeled "SET/USE LISTING XXXX" where XXXX
is the segment name. The table is limited
to 8,000 entries per segment where each
entry represents a <u>reference</u> to a tag, a
location in the direct bank or to the speci-
fic location. When the capacity of the
table is exceeded, a message will appear
at the beginning of the Set/Use listing giving
the card number where the table overflowed.
After outputting the message, program pro-
cessing continues.

<u>2</u> Format

The Set/Use Table listing is divided into
three parts as follows:

Part 1 - This part will contain tags with
core location and card number, and all
references, by card number, to these tags.

Example:

```
DBUFF     4000        14
   25  U      37 S   39 S    41 S
```

Part 2 - This part will contain references,
by card number, to specific locations in
the Direct Bank using Direct and Indirect
addressing modes.

Example:

```
          0017
   50  U      71 S   103U
```

Part 3 - This part will contain references,
by card number, to the specific location
(7777B Bank Ø) using the Specific addressing
mode.

Example:

```
                 7777
      191 S      194 S      267S   272 U   275 S
```

All references to a tag or location will
be followed with a set (S) or used (U)
code.

(2) PASS 2 - BINARY GENERATION

Prior to commencement of PASS 2 the old master is
closed and is not referenced again. Therefore,
it is available for multifiling purposes. In
PASS 2, the new segment (tape), the SST's (disk),
and the CST (core) are combined to generate 160
object code. As the object code is generated, it
is placed into a simulated 160 core bank. If a
"SYMTAPE" of the segment being processed was
requested and File 07 is declared, a card image
tape is generated. Following the generation of the
object binary for each segment, the simulated core
bank is written onto serial disk. Following this
an octal deck on RUN-MOD card format of the object
binary, specifying Bank 2 as the load bank, will
be produced if the user has requested the "OCTDECK"
Control Option and has declared File 06 (card
punch). It is during PASS 2 that the side-by-side
tape is generated. The BADSAP Assembler will
list only those segments changed during PASS 1,
unless the controls LISTALL or BENCHMK have been
used. Errors will be listed in all cases.

(3) PASS 3 - BINARY OUTPUT TAPE PRODUCTION

PASS 3 of the BADSAP Assembly System copies the
interim binary from disc to tape. Prior to each
binary information block a segment identifier is
inserted. Included in this identifier is the first
word address/last word address of the simulated
core bank which follows it. Inasmuch as the new
master tape is closed prior to the commencement
of PASS 3, the binary output of BADSAP (File 05)
may be multifiled to the new master tape (File 02).

b. BADSAP Control

The BADSAP Assembly System has two forms of control,
SPECIFIC and IMPLIED. SPECIFIC controls are inputted
via the card reader (File 03). IMPLIED control repre-
sents the assumptions made by BADSAP when (1) adapting

to an equipment sensitive environment and (2)
functioning without SPECIFIC control input(s).

(1) IMPLIED Control

   (a) GENERATE Mode

      The BADSAP Assembly System normally functions
      in update mode, with various specific control
      inputs governing the update process. It is
      essential that any update system have as an
      alternative a GENERATE mode. This is pro-
      vided in BADSAP. The GENERATE mode of BADSAP
      is activated by omitting the declaration of
      the old master (File 01).

   (b) IMPLIED-ADD

      When BADSAP does not detect recognizable
      specific input defining the update sequence,
      it operates in the IMPLIED-ADD mode. This
      updates the old master (File 01), if declared,
      by placing all symbolic input at the end of
      the new master.

   (c) EDIT LIST Output

      BADSAP provides a complete list of all input
      processed as well as a record of all line-
      entries deleted from the old master when
      operating in the normal (SPECIFIC UPDATE)
      mode. This EDIT LIST is suppressed during
      GENERATE and IMPLIED-ADD runs. In the
      GENERATE or in the IMPLIED-ADD mode, columns
      73-80 of the symbolic input will be placed
      on the master unaltered.

(2) SPECIFIC Control

   All specific control cards are punched starting
   in column 1.

   (a) Update Control

      The cards included in this section must be in
      sequence. The CHANGE cards must be in the
      order that the segments are on the old master.
      The INSERT and DELETE cards which follow each
      CHANGE card must be in numeric sequence
      within the segment.

      NOTE: Each CHANGE card must be immediately
      followed by either an INSERT or a DELETE card.

<u>1</u> CHANGE

The CHANGE card identifies the segment
which is to be changed. The name of the
segment which is to be changed is punched
starting in column 8 of the card.

<u>2</u> DELETE

The parameters in the DELETE card, numbers
specifying symbolic line numbers on the
old master (File 01), are punched starting
in column 8. Parameters on the DELETE
card are separated by a blank. If only
one parameter is specified in the DELETE
card, the single line specified will be
deleted. If two parameters are specified,
the lines from the first parameter to the last
parameter inclusive will be deleted. The
delete card may be followed by symbolic
input, which will be inserted in place of
the deleted instructions.

<u>3</u> INSERT

The INSERT card for BADSAP has one para-
meter, a number specifying the symbolic
line number on the old master (File 01)
after which the code which follows the INSERT
card is to be placed. The parameter is
punched starting in column 8.

(b) Output Control

The BADSAP Assembly System offers three forms
of optional output which can be obtained by
the use of specific control requests.

<u>1</u> OCTDECK

The OCTDECK control card is used to produce
an octal deck of the specified segment of
the object program in RUN-MOD card format.
The segment identification is punched start-
ing in column 9. To assist the user in
identification of each octal deck produced,
the segment identifier will be punched into
the output deck, in columns 77-80. A card
punch (File 06) must be declared, otherwise
the OCTDECK request will be ignored. This
prevents inadvertent abnormal terminations
caused by a missing file card. All octal
decks are punched for BANK 2.

2 SYMTAPE

The SYMTAPE control card in BADSAP is used to extract from the master a symbolic card image tape. The identification of the segment which is to be extracted is punched in the SYMTAPE card starting in column 9. As many segments as are desired may be extracted on the card image tape. SYMTAPE's can be structured as follows:

a If the SEABASS control card is not included, a segment mark will be written after each segment except the last, which will be followed by two segment marks.

b When the SEABASS control card is included, segment marks will **not** be written after each segment. At the end of the last segment requested a card image containing FINIS in card columns 9-13 will be written on the SYMTAPE and will then be followed by two segment marks. This SYMTAPE can then be used as an input symbolic for a SEABASS assembly, which requires a FINIS card for termination of assembly. File 07 must be declared, otherwise the SYMTAPE request will be ignored. This prevents inadvertent abnormal terminations caused by a missing file card.

3 SEABASS

Optional as defined under SYMTAPE above.

c. Side-by-Side List Control

The BADSAP Assembly System will produce a complete side-by-side of all segments during a generate run. During an update run only those lines of a segment which are in error will be listed, unless that segment was changed during that update run, in which case a side-by-side will be produced for the entire segment. For those users familiar with the standard OSAP output the side-by-side produced by BADSAP will represent a slight change. The binary generated by two-word instructions will be listed on a single line. The FDC Pseudo Op in BADSAP representation only requires one line on the side-by-side because the binary for the first word only is listed.

(1) LISTALL

The LISTALL control is used to override the automatic suppression of side-by-side production of those segments that were not changed. Thus the LISTALL control will cause a list of all segments on the master to be produced.

(2) OLDLIST

The OLDLIST control card allows the user to obtain the standard OSAP form of the side-by-side.

(3) BENCHMK

The BENCHMK control to BADSAP alters the symbolic image on the master by placing the four character SEG ID and the generated sequence number in columns 73-80. (The field into which the date is placed during normal updating.) This card will also override the normal suppression of side-by-side output.

(4) SAVDATE

The SAVDATE control is optional during a BENCHMK run if the user desires to save change dates for historical purposes. Change date is normally carried in columns 73-80 until a BENCHMK run is performed at which time the change date is eliminated and replaced by SEG ID and sequence number. Therefore, by using SAVDATE option the change date will be moved to columns 65-72 of the entry in lieu of elimination. However, users should ensure there are no entries containing FDC, BCD, TUBE, DEC, or other Psuedo Ops which have constants stored through column 65 as they would be partially cleared.

d. Special Controls

(1) FLAGBIN

The FLAGBIN control card in BADSAP is provided as an aid in identifying those portions of the program for which binary will not be generated. BADSAP will generate binary code only after it has detected an ORG Pseudo-Op. The generation of binary by BADSAP is discontinued by the detection of the Pseudo-Ops SEG or CON. Under normal BADSAP operation, no indication is given that a set of instructions or constants is not generating binary. This

is the purpose of the FLAGBIN card. The FLAGBIN
card will produce an asterisk (*) on the side-by-
side listing in front of the location field for
those instructions or constants for which binary
was not produced. Additionally, lines of code
thus flagged would be listed as though they were
in error.

(2) The BADSAP Assembly System which is used to assemble
the Atlantic Display System has incorporated within
it a Pseudo-Op not used in standard OSAP. This
Pseudo-Op is TUBE, which functions in a manner
similar to the Pseudo-Op FDC, with which the OSAP
user is familiar. However, the binary generated
by TUBE will be either FDC or internal BCD, depen-
dent upon user specification. If no option is
specified, BCD is assumed.

(a) FDCCODE

The FDCCODE control to BADSAP is used to control
binary generation with a TUBE Pseudo-Op. Its
use forces generation of FDC code.

(b) BCDCODE

The BCDCODE control to BADSAP forces the TUBE
Pseudo-Op to generate internal BCD code.

(3) LIBRARY

The LIBRARY control card in BADSAP gives the user
the capability of calling LIBRARY routines from a
BADSAP master declared as File 13.

(a) If the LIBRARY control card is included and
File 13 is declared, all library routines
will be deleted from the old BADSAP master
(File 01) and all library routines called in
any segment will be added to the new BADSAP
master (File 02) from File 13. This will
allow the user to benefit from changes made
to the library file. Library routines are
called by including the following card any-
where after a SEG card:

card columns	9-11	15-18
contents	LIB	XXXX

where XXXX is the four-character
identification of the library routine.

(b) If a LIBRARY card is included and File 13
is not declared, all library routines will be
deleted from the old BADSAP master (File 01).

See paragraphs 3.8.8o, 3.8.8p, 3.8.8q, and 3.8,8r
for requirements and limitations pertinent to the
use of LIBRARY.

(4) LIBLIST

The inclusion of the control card LIBLIST will
cause called library routines to be listed on the
side-by-side at the end of each segment. Routines
will be listed in the order in which they are called.

e. Coding in BADSAP (OSAP)

(1) Input

Initial input to BADSAP is cards and input des-
criptions which will be given in terms of card
columns; however, in addition to accepting cards
as input, BADSAP will accept magnetic tape input
which contains one card per record in OPCON code.
(Foreign tapes may be used if the end of data is
defined by double segment marks.) All cards input
to the BADSAP Assembly Program which do not contain,
starting in column 1, the previously noted specific
control words will be assumed to be OSAP coding.

(a) Card Format

Columns	Fields
1-6	Location Field
8	Common Label Flag
9-12	Operation Field
*15-72	M-Term
**29-72	Remarks

*   The first blank encountered terminates the M-Term.
** Some Pseudo Operations generate special cases
and are separately documented.

(2) Location Field

The location field (LOC-FIELD) may be blank or may
contain a symbol punched starting in column 1. A
symbol encountered in the LOC-FIELD during assembly
will normally be assigned the current value of the
location counter. Symbols in the location field

will be placed in either or both of two symbol tables. If the symbol is modified by a COMMON FLAG ("C" in column 8), it will be placed in the common symbol table and in the segment symbol table. If the symbol is not flagged, it will be placed in the segment symbol table only. The common symbol table functions as an extension of the individual segment symbol tables. Symbols in the location column must be unique for proper program identification. If location symbols are dupli- cated, an error indication will appear on the side-by-side listing. The symbol name in the location column associated with symbolic lines of coding may be referenced at any time in the M-Field. The character * (asterisk) when used as a one charac- ter symbol will be given the value of the symbolic line in which it appears; therefore, it cannot be defined in the location field. Certain Pseudo instructions require that a symbol be defined prior to use in the M-Field.

(3) Operation Field

The Operation Field (OPC-Field) may be blank or may contain a mnemonic or Pseudo Operation Code. The mnemonic operation codes will be equated to the octal machine instructions. A blank or illegal operation code will be replaced by zero. An illegal operation code will be marked as an error on the side-by-side listing. The assembly will recognize the relative mode of mnemonic opera- tion code instructions. Mnemonics or pseudo-ops start in column 9.

(4) M-Term

The M-Term may be blank, a signed or unsigned symbol, or a signed or unsigned constant; either may be optionally followed by up to four signe' symbols and/or signed constants. The M-Term must start in column 15 and is terminated by the first blank. The M-Term is computed modulo 2^{12}. In cases where the 12-bit M-Term is allowed, the computed value is added to the already computed operation code to make up the entire machine instruction. Some instructions are self-contained in the operation code and the M-Term is ignored. Some instructions allow only for a six-bit E-Term to be added to the previously computed operation code, in which case the computed value of the M-Term that does not also appear in the location column of some line is unassigned and replaced by zero.

3-65

An indication of all unassigned symbols appears
on the side-by-side listing. An illegal M-Term
will be equated to zero. An asterisk (*) is
considered a special symbol and will always be
equated to the address of the symbolic line in
which it appears. A minus zero (-0) is considered
a special symbol and is evaluated as 7777_8.

(5) Remarks

Remarks normally start in column 29. This insures
a degree of downward compatibility with OSAP and
in general produces a more readable side-by-side.
However, remarks may start immediately following
the blank that defines the end of the M-Term. The
remarks field may contain any combination of the
OPCON card characters. The characters may be
placed anywhere in the field and the spacing will
be preserved in the symbolic output. The remarks
field is used in certain Pseudo Operation instruc-
tions for purposes other than remarks.

f. Pseudo Operation Code Summary

All Pseudo Operation Codes are given in the OPC-Field
and cause the assembler to generate an integral number
(possibly zero) of words in the object program. Each
pseudo operation code and the information contained
in the other fields are listed in the symbolic output
at the point they occur in the symbolic input. The
following is the total list of pseudo operation codes
allowable in BADSAP:

(1) SEG

Used to define the identity of each segment on
a BADSAP Master. The SEG identifier is placed in
columns 15-18 (leading blanks in the SEG are illegal).
A SEG pseudo-op sets the ORG and CON location
counters to zero and suppresses the generation of
binary output (enabled by ORG pseudo-op).

(2) ORG

The ORG location counter is given the value of the
M-Field. The LOC-Field is ignored and the M-Term
contains information subject to the same conditions
pertaining to normal machine instructions. Any
symbols appearing in the M-Term must have been
previously defined or an error will result. If
the M-Term of an ORG pseudo operation code is blank,

the code following will be assigned the previous
value of the ORG location counter. This option
may be used in order to resume binary generation
after the use of a CON pseudo instruction. If ORG
does not appear on the first line of coding and
coding starts with the first line, the location
counter will be set to $\emptyset$ and no binary output will
be generated.

(3) ORGR

Same as ORG (BADSAP does not assemble relocatably).

(4) CON

The CON location counter (a separate location counter
for the CON pseudo operation) is given the value of
the M-Term. The LOC-Field is ignored and the M-Term
contains information subject to the same conditions
pertaining to normal machine instructions. Any
symbol appearing in the M-Term must have been
defined previously or an error will result. Lines
of coding (normally EQUs or area definitions in
another bank) following the CON pseudo instruction
will not produce binary until an ORG or ORGR
pseudo instruction is encountered. If the M-Field
of a CON pseudo instruction is blank, the address
giving the following instructions will be the
previous value of the CON location counter. The
value of the CON location counter is not restricted.

(5) BSS

Used to reserve an integral number of storage loca-
tions. The LOC-Field may contain a symbol and the
M-Term contains information subject to the same
conditions pertaining to normal machine instructions.
Any symbol appearing in the M-Term must have been
previously defined. The value of the M-Term gives
the number of storage locations to be reserved. If
the M-Term is zero, one storage location will be
reserved. The location counter is advanced by the
number of storage locations reserved. If a symbol
is given in the LOC-Field, it will refer to the
first storage location reserved. The storage loca-
tions reserved are not set to zero.

(6) BSSZ

Same as BSS.

(7) EQU

Causes a symbol to be defined. The LOC-Field
contains a symbol and the M-Field contains infor-
mation subject to the same conditions pertaining
to normal machine instructions. If the M-Term
contains symbols, they must have previously been
defined or an error will result (the symbol in
the LOC-Field will be flagged as undefined). The
symbol in the LOC-Field is defined as equal to the
value of the M-Term.

(8) EJCT

Causes the line containing the pseudo code EJCT
to be listed in the output and the printer to
eject the paper to the top of the next page. All
other fields are ignored.

(9) REM

Allows for an additional line of remarks to be
inserted in the coding and appear in the symbolic
listing. The LOC-Field is ignored and the DATA-
Field may contain up to 66 characters of remarks.
The entire card is transferred to the output listing
without change.

(10) TUBE

Allows for the insertion of 1 to 28 words of
external BCD or FDC codes into the object program
based upon control option. The LOC-Field may con-
tain a symbol. The first two columns of the DATA-
Field must contain a two digit decimal number,
$k(\emptyset 1 < k < 28)$ and the next $2 \cdot k$ columns contain
characters. The six-bit OCTAL codes for the $2 \cdot k$
characters are stored left to right in k consecutive
storage locations of the object program.

Example:

LOC-Field	OPC-Field	DATA-Field
AC	TUBE	$\emptyset$4EXAMPLE$\triangle$

The above line would cause the object program to
contain:

CONTROL OPTION:	BCDCODE (Normal)	FDCCODE
	External BCD	OPCON
AC		1235
AC+1		0622
AC+2		2521
AC+3		1205

(a) Note that blanks (denoted by "Δ") are counted as characters.

(b) The "FDCCODE" option causes the "TUBE" pseudo op to function as a FDC pseudo op.

(11) FDC

Same as TUBE except FIELDATA code is produced regardless of option.

(12) SPAC

Same as REM.

(13) BNKx

Same as REM. (BADSAP assembles entire segment into one simulated bank.)

(14) LIB

Calls the library routine identified in columns 15-18 from File 13 and causes the called routine to be added to the segment from which it was called.

g. Error Codes

Errors that are detected and printed on the side-by-side listing are:

ERROR CODE	EXPLANATION
U	Undefined symbol in the M-Field. The undefined symbol is evaluated to zero.
N	The number of words (k) to be inserted by the FDC or TUBE pseudo code is not within the limits $0 < k < 28$. K is replaced with 28.

ERROR CODE	EXPLANATION
M	Symbol is defined more than once in the location field.
O	Illegal operation code. Operation code is given a value of zero.
E	E-term of machine instruction is greater than 77.
K	An EQU pseudo op has been used within an active ORG.
C	The flagged line was inserted on this update.
T	Unsuccessful read from the old master (File 01). Processed as REM. Fix by subsequent updating.
*	This line did _not_ produce object code. ("FLAGBIN" used and line not within active ORG.)

NOTE: The appearance of any of the above will cause the line to be listed unconditionally on the side-by-side.

3.8.8   Requirements/Limitations

a. Any attempt to generate a master by declaring a card image tape as File 01 (instead of using File 03 as documented) will cause BADSAP to produce 30 unusable print lines for every card read because of an invalid NENT word. The correct method for generating a BADSAP master is:

(1) Do not declare an old master (File 01).

(2) Input all symbolic input via File 03.

b. The specific BADSAP controls CHANGE, DELETE, and INSERT are not required when operating in the generate mode. In the event that the program inputted does not contain SEG cards, BADSAP will assume a SEG ID of all zeroes. SEG cards may be inserted on a subsequent BADSAP update run. To insert a SEG card at the beginning of such a BADSAP master use the following (XXXX denotes the desired new SEG ID):

```
CHANGE 0000

DELETE 1

    SEG XXXX

    REM
```

c. The maximum number of segments that can be included in
    a single OCTDECK or SYMTAPE card is 1; however, as many
    as 100 SYMTAPE cards or 100 OCTDECK cards may be inputted
    in a given run. Exceeding this number will merely cause
    any cards in excess of this to be ignored. The same is
    true if the number of segment cards in the input exceeds
    100.

d. If more than 100 SEG cards are inputted, any in excess
    of 100 will be ignored. The coding which follows these
    will be assembled under the SEG ID of the last segment.

e. The sequence of the BADSAP control input, with the
    exception of four cards, is not critical in that they
    are merely stored for usage in PASS 2. The exceptions
    included:

        BENCHMK*
        CHANGE
        INSERT
        DELETE

f. BADSAP contains no restrictions requiring unique SEG
    IDs. However, in the event that duplicate SEG IDs
    exist the update sequence is made more complicated
    for the user. The CHANGE cards which control the update
    process will be applied to the next segment on the
    master which matches the name requested to be changed.
    Thus, if necessary to update a segment which follows
    another segment of the same name, a change would have to
    be applied to the initial segment and to an intermediate
    segment. Then, when the desired segment is encountered,
    the appropriate changes may be applied. It should be
    noted, however, that the presence of a CHANGE card by
    itself will not cause a segment to be bypassed. A
    change in symbolic form must be applied against the seg-
    ment with INSERT and/or DELETE controls specified. If
    any unrecognizable BADSAP control is detected prior to
    a CHANGE card all changes specified will be appended
    to the end of the old master because BADSAP, in updating,
    would not be able to match the assumed card SEG ID.

    *The BENCHMK card must precede all CHANGE cards or all
     symbolic input, if it is to be effective throughout
     all of PASS 1.

This would require a subsequent Pass to reinsert
the changes in the proper place as well as requiring
that the changes appended to the last segment of the
tape be deleted.

g. If a DELETE is specified with two parameters and the
second is less than the first, BADSAP will delete one
line of coding and pen an INSERT of the symbolic code
which follows that DELETE card.

h. If a CHANGE card is not followed by a DELETE but is
followed immediately by symbolic code, that code will
be appended to the end of the segment identified in
the CHANGE card.

i. A label in the common symbol table may be defined
only once on an entire BADSAP master. Otherwise, it
will generate a multiple symbol definition error. In
the event that the same symbol is entered into the
common symbol table more than once, it is impossible
to say which would be used during PASS 2 (BINARY
GENERATION). However, if the symbol appears in a
segment as well as in COMPOOL, the segment defini-
tion will override the COMMON definition of the value
attributed to the label in the segment table, and
will be used for the binary generation of that segment
and that segment only. Other segments which did not
specifically define the label would reference the value
of the symbol from the common symbol table.

j. The specific use of the pseudo op "CON" in BADSAP
makes it more difficult to use the D-registers for storage
of constants, in that following a "CON" no binary is
generated. The reason for this is that BADSAP was
specifically written to assemble very large programs
in which the usage of D-registers used for permanent
storage is almost impossible. The synchronization
between various program selections and segments is
maintained via the COMPOOL system.

k. BADSAP may be used with stacked assemblies with a
maximum of 100 segments.

l. There may be a maximum of 2000 symbols per segment.

m. There may be a maximum of 500 symbols in common.

n. BADSAP, unlike the OSAP Assembler, requires that all
LABELS, OPCODES, and M-TERMS start in the first
position of the appropriate field.

o. No library routines can be changed through a BADSAP update. If any changes are attempted, they will be ignored. The only way to change a library routine is to update the library tape.

p. If SEG cards are to be added or deleted in a BADSAP update, the LIBRARY card must not be included; a subsequent BADSAP update must be made to include the library routines. If this restriction is not observed, the integrity of the BADSAP master and resultant binary tape cannot be guaranteed.

q. A maximum of 100 library routines may be called from one segment.

r. If the called library routine includes an imbedded call for a library routine, this routine must be called with a LIB card in the main segment.

s. There may be a maximum of 8000 set/use references per segment.

t. Binaries created using BADSAP must be loaded using either BADSAPLOADER or a special bootstrap.

3.8.9   Peripheral Equipment

a. A BADSAP run to generate a master would require, in the minimum configuration, two tape drives (File 15 multifiled to File 04; File 05 multifiled to File 02), one disk module (Files 11 and 12, both multifiled to File 10), and one card reader (File 03).

b. A BADSAP run to update a master would require, in the normal configuration, three tape drives (File 15 multi-filed to File 04; File 07 multifiled to File 01; File 05 multifiled to File 02), one disk module (Files 11 and 12 multifiled to File 10), and one card reader (File 03).

c. Additional configurations may be obtained in either a generate or an update run by:

   (1) Adding a card punch (File 06) when an actual output deck is desired.

   (2) Replacing the card reader (File 03) with a card image magnetic tape as input.

   (3) Declaring File 13 when it is desired to utilize library routines.

3-73

3.9 ABCS (CODAP ASSEMBLY PROGRAM)

3.9.1 Program Summary

3.9.1.1   Language: 1604 Assembly language.

3.9.1.2   Function: To maintain, assemble, and test 1604 CODAP programs.

3.9.1.3   Options: See paragraph 3.9.2

3.9.1.4   Entry to Program: Standard SSEC/MCSIM control cards.

3.9.1.5   Requirements/Limitations: See paragraph 3.9.2.

3.9.1.6   Linkage: See paragraph 3.9.2.

3.9.1.7   Printouts: No operational printouts.

3.9.1.8   Method: See paragraph 3.9.2.

3.9.1.9   Subroutines Required: Not applicable.

3.9.1.10   Tables and Files: Varies by function. See paragraph 3.9.2.

3.9.1.11   Peripheral Equipment: Magnetic tape and disk.

3.9.1.12   Storage Locations

Use core from 11000 to 77776. However, the user may utilize location 11000 through 72777 if the assemble function is not needed or after all assemblies have been completed.

3.9.1.13   Sample Task Deck: See paragraph 3.9.4.

3.9.2 Program Description

3.9.2.1   General Description

ABCS contains PLOP, UPDATE, and ASSEMBLE (modified NAVASSEMBLER). Each is explained below.

PLOP is the main control within ABCS. Each function is called into action by a control card, contained in file 10 (the card file).

PLOP control cards are semi-free form, main control in column 1. Secondary control items are separated by blanks in all cards except patch cards with repeated full octal fields. These are separated by commas, the last one being terminated by a blank.

Paragraph 3.9.5 is an example of the manner in which ABCS is used.

3.9.2.2   Main Control

A   -   Assemble
D   -   Dump Core
X   -   Exercise I/O
L   -   Load a program
M   -   Update (Maintain) a symbolic "deck"
P   -   Print a line
U   -   Unload a program
W   -   Wait for completion of I/O
X   -   Execute a program or subroutine
—   -   All others, make a patch

Assemble

Card form and mnemonic op codes are the same as for PPS III and
NAVASSEMBLER. Pseudo ops are as follows:

FINIS	FLEX	BES
IDENT	TEL	BSSZ
END	TYP	BSS
EJECT	FDC	REPEAT
SPACES	EQU	ENDS
REM	OCT	SET
ORG	DEC	
ORGR	BESZ	

The HED pseudo op has been removed.

ENDS has the same meaning and use as END.

REPEAT (with a number (n) in the M field) operates as if the following
card appeared n + 1 times ($1 \leq n \leq 50$).

SET (with an octal value in the m field and a symbol in the loc field)
equates the symbol as an operation with the given octal value. If
a symbol is set more than once, it is defined from the first usage
and all subsequent definitions are ignored. No more than 10 items may
be set in any given segment (as defined by an END card).

Special Features:

    a. The input to the assembler must be preprocessed
        through UPDATE to place it on disc (File 11).

    b. All Tape and Disc files are blocked at 350 words/record.

    c. Approximately 1150 cards are held in core before use
        is made of file 14 (Scratch Disc), so that "small"
        decks do not require the definition of file 14.

<u>Files Used</u>:

```
01   ABCS Program file
04   SXS listing (option 12)
11   Blocked symbolic input (Random disc)
13   Binary Output
14   Scratch Random disc (if required)
10   Card input
15   Old master source
16   New master source
```

<u>Logic</u>:

The assembler processes segments (a group of cards headed by an IDENT card and terminated by an END card) until it detects an end-of-file condition or a FINIS card. The FINIS card is not necessary. It is only included for compatibility reasons.

Dump Core

Dump core from FWA through LWA on the ABCS list file (file 04). Columns 3-40 of the D card contain the FWA and LWA values (octal form).

Exercise I/O

Set up a call to INOUT. Columns 3-40 of the I card contain FC (decimal), FCN (octal), FWA (octal), LWA (octal), and T (decimal). T (track) is only necessary for Random disc I/O.

Load

Columns 3-40 of the L card contain INCREM (octal), PRGID (OPCON), PRGFL (decimal). INCREM is the relocation increment (if needed). PRGID is the program name and PRGFL is the file to be used (file 01 if not supplied).

Update

Normal operation is blocked input, blocked output, notify the user if there are card sequence errors, normal update and resequence output records. These options will prevail if columns 9-13 of the M card are blank or contain anything other than zero. Any or all of these standard options may be changed as follows:

M Card

```
Column   9, 0 = unblocked input (File 15)
        10, 0 = unblocked output (File 16)
        11, 0 = ignore card sequence errors.
        12, 0 = simulate card-to-tape (no file 15)
        13, 0 = do not resequence
```

Files Used:

    01 ABCS Program file
    04 History (PRINTITALL option 12 format)
    10 Card reader
    11 Selected symbolic output (blocked, scratch random
        disc) (See special feature 1.)
    15 Symbolic input (blocked if column 9 of the M card
        is not 0, not used if column 12 is 0).
    16 Symbolic output (blocked if column 10 of the M card
        is not 0).

Special Features:

a. All segments specified by an IDENT card that is replac-
ing another IDENT card will be written on file 11 for
subsequent assembling. Thus, to assemble an entire
card image tape which contains several segments a new
IDENT card must be inserted for each segment. The
complete updated card image file will be written on
file 16.

b. Insertions and replacements are controlled by the se-
quence field of the card deck (file 10) and the symbolic
input file (file 15). This field consists of columns
73-80 of the card image. If the sequence field of a
file 10 item equals the sequence field of a file 15
item, the file 10 item replaces the file 15 item on
file 11 and/or file 16. If the sequence field of a
file 10 item does not match a file 15 item, it will
be inserted in front of the first file 15 item with
a larger value. The sequence field may contain any
OPCON character.

c. Deletions are controlled with special delete cards.
These cards have a 9 in column 1 and two sequence
fields. The first sequence field consists of columns
9-16 and the second of 17-24. The deletion takes place
from the sequence value of field one through the
sequence value of field two. If field two is blank,
the single card defined by field one will be deleted.

d. An end-of-file on fi' 10 may be simulated by the use
of a card with 9END in columns 1-4. This is necessary
if any PLOP control cards are to appear behind the
UPDATE deck.

e. A special merge action is available if it is desired
to place the contents of an unblocked file at the
current position of file 11 and/or 16. A special
control card with 9TAP in columns 1-4 and a file code
(decimal) in columns 9-10 will cause the UPDATE

program to open the file specified, transfer the
contents to file 11, and/or file 16, close the file,
and read the next card. If the merging is being done
in the middle of a segment that has been selected for
file 11, then the 'merged' cards will also be trans-
ferred to file 11. Otherwise, they will only appear
on file 16 (this is the more normal case).

    f. Each output image may be resequenced, starting with
1000_8 and increasing by 1000_8 for each card (controlled
by column 13 of the M card).

Print

Columns 3-40 of the P card contain LOC (octal). This is the origin
of a 15 word line that is to be inserted in the blocked print tape
(file 4).

Unload

Generate a program file that can be loaded by the system. Columns
3-40 of the U card contain INCREM (octal), LWA (octal), PRGID (OPCON),
PRGFL (decimal).

INCREM is the first location, LWA is the last location + 1, PRGID is
the program name and PRGFL is the file to be used.

A program file is generated with the proper control blocks and one
data block containing the information from INCREM to LWA.

Wait

Columns 3-40 may contain FC (decimal) or be blank. If blank, the wait
function will wait until location 70_8 becomes negative (all I/O quiet).
If a file code is present, it will wait until the file specified
becomes _not_ busy.

Execute

Column 2 contains R or is blank. R signals to transfer control by an
RTJ, blank by SLJ. Columns 4-40 contain TRAD (octal). This is the
location to which control is to be transferred.

Patch

Columns 2-6 may contain a location (octal), column 7 may be blank or
contain U or L. If L, the patch will be made to the lower instruction;
otherwise, it will be made to the upper.

Columns 9-40 is a free field containing a symbolic OP and address
value or a b value and an address value (both octal or the address of

the form * ± octal). The symbolic OPs consist of all the symbolic
instructions defined in CDC 303, as well as, RTJ, NOP, and REM.

Or the following: Columns 9-40 is a free field containing octal
values separated by commas (each will be treated as a full word).

Any patch card with a R in column 1 will have the value in INCREM
added to the location and/or address values. This allows the user
to make relocatable patches to a program or routine previously loaded
by a LOAD function.

3.9.2.3   PLOP Logic

    a. PLOP opens file 10 for Input

    b. Then opens file 04 for Output

    c. Next, it reads a card. If EOF, close files and exit.
       If not, process according to column 1 and return to 3.

Notes

LOC 76000 is logically step c above, so a useful method is to write
routines with PATCH cards and have each routine SLJ 76000 when it is
done. Then with X cards, execute them in desired sequence. Dump
between executions if you want to.

There are some conversion routines within PLOP that your routine
can use. See the latest listing for linkage.

File 4 is always open for output, so your routines can write on it
if desired, but it is blocked, so that care must be exercised in its
use (print lines may appear out of order with other lines).

Similarly, file 10 is always open for input, but you have to be care-
ful about having your routine read the next card. All numbers in
PATCH card are OCTAL.

For example:

          LDA*+9 is invalid
          AJP M is invalid
          QJP 3 is okay
          ENA*+50 is okay (50_8 that is)

<u>System Symbols Defined by ABCS, Assemble Function</u>

<u>Symbol</u>	<u>Definition</u>
INPUT	154B
OUTPUT	155B
INOUT	142B
NORMAL	156B
RVLDR	147B
WAIT	143B
DUMP	153B
FAULT	152B
ICW	56B
ITA	70B
S15	142B
S10	143B
S9B	147B
S21	155B
U03	172B
U04	173B
U07	176B
U08	177B
S30	202B
S02	204B
S01	205B
S04	211B
S37	215B
S38	216B
T5A	221B
F01	222B
U20	224B
U22	226B
IOM	250B
ABORTA	300B
ABORTB	301B
ABORTC	302B
ABORTD	303B
ABORTE	304B
TINF	320B
SDFILNO	1035B
LIMITS	1056B
CW1	56B
CW2	57B
CW3	60B
CW4	61B
CW5	62B
CW6	63B
CW7	64B
CW8	65B
CW9	66B

3.9.4 Required MCSIM Control Cards

See NAVCOSSACT Document No. 12M901D TR-01 (FOCCPAC 1604-A and 160-A SSEC/MCSIM User's Guide), Appendix H and the FOCCPAC Programmer's Information Notebook (PIN) for a detailed description of each control card.

3.9.4.1 Time Card

Columns	Contents	Explanation
1	A	Identifies card as a time card.
3-4		Year
5-6		Month
7-8		Day
9-10		Hour
11-12		Minute

Columns 3-12 may be left blank if no DTG information is desired in output file labels.

3.9.4.2 Job Card

Columns	Contents	Explanation
1	J	Identifies card as a job card.
3-4		Precedence code. See PIN, section 4.3 for details.
5		Security classification: U - Unclassified C - Confidential S - Secret T - Top Secret O - SIOP E - Sensitive I - Super Sensitive
7-14		Job identification
15-22		FOCCPAC program number
23-28		Programmer's name
29-30		Programmer's number
31-32		Number of tasks

3.9.4.3 Task Card

Columns	Contents	Explanation
1	T	Identifies card as a task card.
3-4		Task number - must be a numeric value from 01-99.
5-12		Task ID.
25	1	Indicates that this is a 1604 task.
26-28		Estimated running time. Column 26 indicates hours (0-9); columns 27-28 indicate minutes.

3.9.4.4 File Card (01)

Columns	Contents	Explanation
1	F	Identifies card as a file card.
3-4	01	File code for program file.
5-6	MT	Device assigned for this file (magnetic tape).
7	P	Use (protected input).
9	U	Security classification (unclassified)
20	P	Preload indicator.
27-40	ABCS	File ID.

3.9.4.5 File Card (04)

Columns	Contents	Explanation
1	F	Identifies card as a file card.
3-4	04	File code for History file.
5-6	MT	Device assigned for this file (magnetic tape)
7	O	Use (output)

Columns	Contents	Explanation
9		Security classification: U - Unclassified C - Confidential S - Secret T - Top Secret O - SIOP E - Sensitive I - Super sensitive
27-40	SXS/HISTORY	File ID
49-50	12	PRINTITALL option for this tape format (blocked, with print format control characters).

3.9.4.6   File Card (10)

Columns	Contents	Explanation
1	F	Identifies card as a file card.
3-4	10	File code for card file.
5-6	CR	Device assigned for this file (card reader).
7	I	Use (input)
9		Security classification: U - Unclassified C - Confidential S - Secret T - Top Secret O - SIOP E - Sensitive I - Super sensitive
27-40	CARDS	File ID

3.9.5   Sample Deck Setup

Assume that the user wishes to update a card image tape, assemble it, load it, patch in a small check-out procedure, and execute it. A dump will be taken at termination.

```
            A
            J
            T
            F 01MTP U      P        ABCS
            F 04MTO U               SXS/HISTORY     12
            F 10CRI U               CARDS
            F 11DRS U               ASSEMBLERINPUT
            F 13MTO U               BINARYOUTPUT
(Note: )    F 14DRS U               SCRATCH
            F 15MTP U               OLDCARDIMAGE
            F 16MTO U               NEWCARDIMAGE


                   Blank

            ........MCARDS

            M              (Update control card)

            .    IDENT TST1 ⎫
            .               ⎪
            .               ⎬  Modification Cards
            .               ⎪
                            ⎭

            9END           (Simulated end-of-file card)
            A              (Assemble control card)
            L    TST1  13. (Load control card)
            .            ⎫
            .            ⎬ Patch Cards
            .            ⎭
            X    20000     (Execute control card)
            D    15000  30000  (Dump control card)
            ........  ECARDS
```

(Note: File 14 may be multifiled to File 11. Specify nonoverlapping cylinder limits.)

3.10   GENERALIZED CODE CONVERSION

3.10.1   Program Description

3.10.1.1   Language: JOVIAL and Assembly language.

3.10.1.2   Function

A 1604A JOVIAL and Assembly language program to convert data on mag-
netic tape between any two standard data interchange codes. Codes
supported are:

           1) OPCON
           2) Binary coded decimal
           3) American Standard code for information interchange
           4) Extended Binary coded decimal interchange code
           5) WWMCCS subset of BCD

Data may be recorded in odd or even parity, 200 or 556 BPI, 7 channel
NRZI magnetic tape. Data is read from magnetic tape and converted
according to parameter submitted on a control card. Input records
are limited to 1000 1604 words in length. Any records longer than
1000 1604 words will be truncated. Conversion from 6-bit to 6-bit
codes is quite rapid with an 80 character record requiring approxi-
mately 2 milliseconds. 6-bit to 8-bit code conversions are slower
requiring 6 MS per 80 characters record. The time increase is due to
the repacking necessary since only six (6) 8-bit characters can be
stored in one (1) 1604 word.

3.10.1.3   Options

           a. Input tape may be odd or even parity. Program assumes
               odd parity if none is included in the parameter card.

           b. A number of segment marks may be skipped before
               conversion takes place.

           c. A specified number of segments may be converted.

           d. The following codes are available:

               1) OPCON code
               2) External BCD code
               3) Internal BCD code
               4) American Standard code for information inter-
                   change (ASCII)
               5) Extended Binary coded decimal interchange
                   code (EBCDIC)
               6) WWMCCS subset of BCD

3.10.1.4   Entry to Program

Entry to convert is made by using standard system control cards.

3.10.1.5   Linkage

    a.   Input

        Files 01-27 are available to the user.
        File 29 - contains the parameter card, one card for
            each file to be converted.
        File 30 - contains the binary program and is preloaded
            from the ACE master disk.

    b.   Output

        Files 01-27 are available to the user.
        File 28 - will contain the history of events occurring
            during the conversion run.

3.10.1.6   Sample Task Deck

    a.   Control Cards

        (1)   Time, Job and Task cards are prepared in accordance
        with instruction in Appendix H of NAVCOSSACT Docu-
        ment No. 12M901D TR-01.

        (2)   File Cards

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	01*	File code of first tape to be converted. * Could be 01 through 27.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U, C, S, T, O, E or I	Security classification of this file.
26	H or L	Denotes density of foreign tape.
26	S or Blank H or L	S or Blank system tape, L=low density foreign tape, H=high density.

27-40		File identification name of tape to be foreign converted.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	29	File code of parameter file.
5-6	CR	Device type (card reader) assigned this file.
7	I	Use (input) of this file.
9	U	Security classification of this tape.
26	S or Blank	S=system file.
27-40		PARAMCARDS (or any suitable file name).

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies cards as a file card.
3-4	30	File code of this utility program.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	P	Use (protected input) of this file.
9	U	Security classification of this file.
15-16	MM	Indicates that this file is multifiled to the SSEC master.
20	P	Indicates that first segment of this file is to be preloaded.
26	S or Blank	System file.
27-40	CONV	File identification name of program file.

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies cards as a file card.
3-4	28	File code of this print file.
5-6	PR	Device type (printer) assigned this file.
7	S	Use (scratch) of this file.
9	U	Security classification of this file.
15-16	MM	Indicates that this file is multifiled to the SSEC master.
27-40		File identification (any suitable file name).
44	N	Print the file regardless of the type of termination.
49-50	03	Print Option

COLUMNS	CONTENTS	EXPLANATION
1	F	Identifies card as a file card.
3-4	02*	File code of output tape * 01-27 available to user.
5-6	MT	Device type (magnetic tape) assigned to this file.
7	O	Use (output) of this file.
9	U, C, S, T, O, E or I	Security classification of output file.
26	S or Blank H or L	S or blank system output, H=high density foreign, L=low density foreign.
27-40		Name of output file.

b. Parameter Cards

One parameter card is required for each file to be
converted. Tape may be declared in system or foreign
format. Parameters may be placed on the card in free

form. At least one space must separate parameters.
The only restriction is that input description follows
the input statement and output description follows
the output statement.

PARAMETER		MEANING
Input	(required)	Input file description follows.
*FN	(required)	Asterisk precedes 2 digit file number of input file.
Parity	(optional)	ODD for odd parity, EVEN for even parity. ODD parity is assumed if omitted.
Code	(optional)	OPCON = OPCON code
		XBCD = External BCD code
		IBCD = Internal BCD code
		ASCII = American Standard code for information interchange
		EBCDIC = Extended binary coded decimal interchange code.
		WWMCCS = WWMCCS subset of BCD or WIMMIX = WWMCCS subset of BCD
		<u>If omitted OPCON is assumed!</u>
Output	(required)	Output file description follows.
*FN	(required)	Asterisk precedes the 2 digit file number of output file.
Parity	(optional)	ODD for odd parity. EVEN for even parity. ODD is assumed if omitted.
Code	(optional)	Same as code for input.
Skip XX	(optional)	Number of segment marks to be skipped on the input tape before conversion begins. If omitted none is assumed.
Convert XX	(optional)	Number of segments to be converted from the input tape.

$ (required)   Terminator. Comments may follow
                 $ on the parameter card.

Example of parameter card.

    INPUT   *01 OUTPUT   *17 XBCD EVEN CONVERT 2 $

The above control card indicates that the input file is
ODD parity, OPCON code and that the output is to be in
EVEN parity External BCD code. Conversion will halt
after two (2) segment marks have been read from the
input tape. Also, no segment marks were skipped.

3.10.2   Sample File Cards

```
........          EPARAMCARDS
INPUT *01 OPCON ODD OUTPUT *02 XBCD EVEN $          OPCON TO BCD CONVERSION
........          MPARAMCARDS
                              SENTINEL CARD
F  02MTO U         HOUTPUT/FILE///          30
TF 01MTP B         SINPUT/FILE//92
TF 28PRS U    MM   SCONVERT/LIST92   N   03
TF 29CRI U         SPARAMCARDS
TF 30MTP U    MM  P  SCONV
D1CONVERT          102030
J 770 JOBIDENTF93/9992UUEP1DSC01
```

3.10.3   Code Conversion Tables

OCTAL CODE	GRAPHIC CODE	XBCD CODE	XBCD GRAPHIC	IPCD CODE	IPCD GRAPHIC	ASCII CODE	ASCII GRAPHIC	EBCDIC CODE	EBCDIC GRAPHIC	USASCCS CODE	USASCCS GRAPHIC
00	7 .	20	Space	60	Space	050	"	154	~	74	~
01	↑	57	Δ	57	Δ	040	Space	100	Space	40	↑
02	↓	37	⫤	77	⫤	040	Space	100	Space	72	←
03	∫	75		35		040	Space	100	Space	72	
04	]	55	]	55	]	040	Space	100	Space	34	]
05	Space	20	Space	60	Space	040	Space	100	Space	70	Space
06	A	61	A	21	A	101	A	301	A	21	A
07	B	62	B	22	B	102	B	302	B	22	B
10	C	63	C	23	C	103	C	303	C	23	C
11	D	64	D	24	D	104	D	304	D	24	D
12	E	65	E	25	E	105	E	305	E	25	E
13	F	66	F	26	F	106	F	306	F	26	F
14	G	67	G	27	G	107	G	307	G	27	G
15	H	70	H	30	H	110	H	310	H	30	H
16	I	71	I	31	I	111	I	311	I	31	I
17	J	41	J	41	J	112	J	321	J	41	J
20	K	42	K	42	K	113	K	322	K	42	K
21	L	43	L	43	L	114	L	323	L	43	L
22	M	44	M	44	M	115	M	324	M	44	M
23	N	45	N	45	N	116	N	325	N	45	N
24	O	46	O	46	O	117	O	326	O	46	O
25	P	47	P	47	P	120	P	327	P	47	P
26	Q	50	Q	50	Q	121	Q	330	Q	53	Q
27	R	51	R	51	R	122	R	331	R	51	R
30	S	22	S	62	S	123	S	342	S	62	S
31	T	23	T	63	T	124	T	343	T	63	T
32	U	24	U	64	U	125	U	344	U	64	U
33	V	25	V	65	V	126	V	345	V	65	V
34	W	26	W	66	W	127	W	346	W	66	W
35	X	27	X	67	X	130	X	347	X	67	X
36	Y	30	Y	70	Y	131	Y	350	Y	70	Y
37	Z	31	Z	71	Z	132	Z	351	Z	71	Z
40	)	74	)	34	)	51	)	135	)	55	)
41	-	40	-	40	-	255	-	140	-	52	-
42	+	60	+	20	+	253	+	116	+	60	+
43	<	76	<	36	<	274	<	114	<	36	<
44	=	13	=	13	=	075	=	176	=	75	=
45	>	16	>	16	>	276	>	156	>	16	>
46	Prime	14	Prime	14	Prime	052	Prime	175	Prime	57	Prime
47	$	53	$	53	$	244	$	133	$	53	$
50	*	54	*	54	*	052	*	134	*	54	*
51	⊂	34	⊂	74	⊂	250	⊂	115	⊂	35	⊂
52	≥	35	∨	35	∨	245	?	100	Space	76	?
53	Colon	15	Colon	15	Colon	272	Colon	100	Space	15	Colon
54	?	72	?	32	?	277	?	100	Space	17	?
55	!	52	!	52	!	241	!	100	Space	77	!
56	Comma	33	Comma	73	Comma	054	Comma	153	Comma	73	Comma
57	≠	32	≠	72	≠	040	Space	100	Space	33	≠
60	↑	12	↑	00	↑	260	?	360	?	00	?
61	1	01	1	01	1	261	1	361	1	01	1
62	2	02	2	02	2	262	2	362	2	02	2
63	3	03	3	03	3	263	3	363	3	03	3
64	4	04	4	04	4	264	4	364	4	04	4
65	5	05	5	05	5	265	5	365	5	05	5
66	6	06	6	06	6	266	6	366	6	06	6
67	7	07	7	07	7	267	7	367	7	07	7
70	8	10	8	10	8	070	8	370	8	10	8
71	9	11	9	11	9	271	9	371	9	11	9
72	∧	36	\	76	\	334	\	100	Space	37	\
73	;	56	;	56	;	073	;	100	Space	56	;
74	/	71	/	61	/	057	/	141	/	61	/
75	.	73	.	33	.	256	.	100	Space	23	.
76	→	20	Blank	60	Blank	040	Space	100	Space	32	⊄
77	⇒	77	≢	37		040	Space	100	Space	14	∩

SECTION IV

1604A NONSYSTEM UTILITIES

4.1.1

None available at this time.

APPENDIX A

SUMMARY OF 160A NONSYSTEM UTILITY PROGRAMS

$A-1.1$

APPENDIX A

SUMMARY OF 160A NONSYSTEM UTILITY PROGRAMS

A.1   SYSTEM UTILITIES

See FOCCPAC Document No. F12S122 TR-51.

A.2   NONSYSTEM UTILITIES LISTED BY FILE SEGMENT NUMBER

1. Nonsystem utility bootstrap and loader segment

   (Segment 0000 and 0067)
   To call all 160A nonsystem utilities and maintenance segments from the program module of the ACE disk system or magnetic tape.

2. Bi-octal paper tape punch

   (Segment 0001)
   This routine will dump memory in octal machine load format on seven level paper tape.

3. CINCPACFLT/FOCCPAC data link

   (Segment 0002)
   To receive or transmit data via the 8066 data link (FOCCPAC side).

4. CINCPACFLT/FOCCPAC data link

   (Segment 0003)
   To receive or transmit data via the 8066 data link (CINCPACFLT side).

5. AUTODIN card to OPCON cards

   (Segment 0004)
   To convert AUTODIN coded cards to OPCON coded cards so they may be used within the FOCCPAC system.

6. Paper tape to magnetic tape or open-ended file

   (Segment 0005)
   To prestore paper tape on magnetic tape or open-ended file.

7. Magnetic tape to cards

   (Segment 0006)
   To punch cards from magnetic tape.

8. Card to print 80/80 list

   (Segment 0007)
   Routine prints card either single or double space.

9. Unassigned

   (Segment 0010)

10. Core dump on magnetic tape or printer

    (Segment 0011)
    To get a core dump of the 160A for debugging a program.

11. Unassigned   (Segment 0012)

12. Paper tape to mylar tape   (Segment 0013)
    copier   To be used in the event the tally
    is down for copying paper tape to mylar
    tape.

13. Unassigned   (Segment 0014)

14. Dump disk on the   (Segment 0015)
    printer   Dump any disk on the printer.

15. Unassigned   (Segment 0016)

16. TELMAC prestore   (Segment 0017)
    Prestore TELMAC tape to open-ended
    file.

17. Unassigned   (Segments 0020 through 0034)

18. DFM directory rebuild   (Segment 0035)
    Allows the maintenance of the DFM
    directory--this is a subprogram of
    the system utility DFM.

19. PCC   (Segment 0036)
    Provides a means of maintaining
    permanent control cards on the ACE
    system disk.

20. Mnemonic dump   (Segment 0037)
    To dump core in mnemonic for on the
    line printer.

21. Unassigned   (Segment 0040, 0041)

22. 2 bank printitall   (Segment 0042)
    A 2 bank tape print.

23. Unassigned   (Segment 0043)

24. Master disk rebuilder   (Segment 0044)
    To be used to rebuild the master disk
    in the event the system was destroyed.

25. Unassigned   (Segments 0045 through 0053)

26. 160A tape labeler   (Segment 0054)
    A means of putting system labels on a
    magnetic tape via the 161 typewriter
    and using a 160A.

27. Unassigned   (Segment 0055)

28. BCD tape print   (Segment 0056)
                                        Print BCD (even parity) tapes.

29. Unassigned   (Segments 0057 through 0065)

30. 160A SSEC bootstrap   (Segment 0066)
                                        It is used to call the program off the
                                        SSEC tape.

31. NSU bootstrap   (Segment 0067)
                                        To load enough instruction to call
                                        segment 0000 in core.

32. Unassigned   (Segments 0070 through 0073)

33. 8090 monitor   (Segment 0074)
                                        To provide an interface between the
                                        8090 and ACE Control using the 161
                                        typewriter.

34. Grasp specification   (Segment 0075)
    compiler   To compile OCR page reading specifica-
                                        tion on tape in the format expected
                                        by segment 0076.

35. Grasp prestore   (Segment 0076)
                                        To read OCR pages, pull the specs from
                                        disk, and prestore the text to an
                                        open-ended file.

36. Unassigned   (Segment 0077)

APPENDIX B

SUMMARY OF 1604A SYSTEM UTILITY PROGRAMS

B-1.1

APPENDIX B

SUMMARY OF 1604A SYSTEM UTILITY PROGRAMS

B.1   SYSTEM UTILITIES

1. JOT

    (File Segment ID: JOT)
    A program that converts data definition
    found in an IPS Master File Format
    record to a format acceptable to the
    JOVIAL compiler or the Compool assembly
    program.

2. NAVASSEMBLER

    (File Segment ID: ASM)
    A program that assembles any number of
    prestored CODAP coded programs that
    are non-Compool sensitive into loader
    control binary format.

3. SNOCTALF

    (File Segment ID: SNCT)
    A program that provides the capability
    of utilizing an unlimited number of
    octal correction and/or up to 20
    distinct snapshot core dump cards
    per computer run.

4. SYSTRACE

    (File Segment ID: SYST)
    A program that provides services to the
    user to locate and diagnose program
    errors as well as a means of correcting
    them with a minimum of programming
    effort.

5. TAPECONVERTBIN

    (File Segment ID: FORN)
    A program that converts BCD or Binary
    Records from an IBM 1401, 1410, 704
    or 7090 created magnetic tape to OPCON
    coded records written as a file on
    an OPCON system labeled magnetic tape.

6. UNLIMITED OCTAL
    CORRECTIONS

    A seven octal correction card program
    that provides for an unlimited number
    of system formatted octals to be
    read into the user's program.

7. ECOP

    (File Segment ID: ECOP)
    A program to copy foreign even parity
    magnetic tapes and convert any 00 codes
    to 77 codes to allow processing in the
    OPCONCTR.

8. BADSAP ASSEMBLER

(File Segment ID: BADS)
BADSAP is a three-pass assembly program
which converts programs written in
symbolic notation (BADSAP/OSAP) to
a fixed 160A machine language program.

9. ABCS

(File Segment ID: ABCS)
Assembles, tests, and maintains
1604A CODAP programs.

10. GENERALIZED CODE
CONVERSION

(File Segment ID: CONV)
Converts data on magnetic tape between
any two standard data interchange codes.

B.2   NONSYSTEM UTILITIES

None available.