50272-101

REPORT DOCUMENTATION PAGE	1. REPORT NO. DOD/DF-81/010b	2.	3. Recipient's Accession No.
4. Title and Subtitle: HONEYWELL H6000 UTILITY SOFTWARE SYSTEM, Vol II, User's Manual			5. Report Date: December 1976
7. Author(s)			8. Performing Organization Rept. No.
9. Performing Organization Name and Address: U.S. Department of the Air Force, Headquarters, Washington, D.C. 20330			10. Project/Task/Work Unit No. 11. Contract(C) or Grant(G) No. (C) (G)
12. Sponsoring Organization Name and Address: (same)			13. Type of Report & Period Covered 14.

15. Supplementary Notes

For magnetic tape, see AD. A105692

16. Abstract (Limit: 200 words)

The Honeywell H6000 Utility Software System consists of a composite of many different types of basic (non-functional) software required in support of H6000 functional automated data systems. This software includes utility and general purpose programs/subroutines which perform specific functions common to many users. Code conversion, data compaction, and tape file input/output are a few of the functions performed by the system.

17. Document Analysis a. Descriptors

Utilities
Data compaction
Code conversion

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement: The source agency has restricted sales of this item to Federal, state and local governments.	19. Security Class (This Report): UNCLASSIFIED	21. No. of Pages: 116
	20. Security Class (This Page): UNCLASSIFIED	22. Price

(See ANSI-Z39.18) See Instructions on Reverse OPTIONAL FORM 272 (4-77) (Formerly NTIS-35) Department of Commerce

81 9 17 .53

REPRINT 24 April 1979
(includes C1 thru C7)

DEPARTMENT OF THE AIR FORCE
Headquarters U[illegible] Air Force
Washington DC 20330

AFM 171-[illegible]
Volume II
1 December [illegible]

Automatic Data Processing Systems and Procedures

H6000 UTILITY SOFTWARE USERS MANUAL: P891/ZA

This manual provides a computer information for utilization of the systems, programs, and subroutines of H6000 standard Air Force utility software. This manual is the central point of documentation for all standard Air Force H6000 utility software.

USERS MANUAL
TABLE OF CONTENTS

Supersedes AFM 171-604, Volume II, 27 April 1976. (For summary of revised, deleted, or added material, see signature page).
OPR: AFDSDC/SDM (by delegation)
DISTRIBUTION: F

SECTION 8. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

PART FIVE - GENERAL PURPOSE TAPE FILE INPUT UTILITY

SECTION 9. SYSTEM SUMMARY

PART TEN - COBOL AIDS

LIST OF FIGURES

DEPARTMENT OF THE AIR FORCE
Headquarters US Air Force
Washington DC 20330

CHANGE 8
AFM 171-604
Volume II
1 November 1979

Automatic Data Processing Systems and Procedures

H6000 UTILITY SOFTWARE USERS MANUAL: S891/ZA

AFM 171-604, Volume II, 1 December 1976, is changed as follows:

Write-In Changes:

Page	Reference	Action
4-1	Para 4.1.1.1f	Change "PCNSP891P11" to "PCNSS891P11."
4-2	Para 4.2.1.1d	Change "PCNSP891P21" to "PCNSS891P21."
8-2	Para 8.1.2.4a	Change "PCNSP891A00" to "PCNSS891A00."
8-3	Para 8.1.2.4b	Change "PCNSP891A01" to "PCNSS891A01."
8-6	Fig 8-01	Change "SP891-A00" to "SS891-A00."
		Change "PCNSP891A00" to "PCNSS891A00."

No. of Printed Pages: 13
DISTRIBUTION: F

8-7	Fig 8-02	Change "SP891-A00" to "SS891-A00."
8-8	Fig 8-03	Change "SP891-A01" to "SS891-A01."
		Change "PCNSP891A01" to "PCNSS891A01."
8-9	Fig 8-04	Change "SP891-A01" to "SS891-A01."
8-10	Fig 8-05	Change "SP891-A01" to "SS891-A01."
8-11	Fig 8-06	Change "SP891-B01" to "SS891-B01."
		Change "PCNSP891B01" to "PCNSS891B01."
8-12	Fig 8-07	Change "SP891-B01" to "SS891-B01."
10-3	Para 10.2.2.1	Add "MTH4ØØ Tape Unit" below "single file."
10-4	Fig 10-01	Change "PCNSP891T11" to "PCNSS891T11."
10-8	Fig 10-03	Change "PCNSP891T21" to "PCNSS891T21."
12-2	Para 12.1.2.1	Change "PCNSP891K11" to "PCNSS891K11."
	Para 12.1.2.4	Change "PCNSP891K11" to "PCNSS891K11."
14-1	Para 14.1.2.1a	Change "PCNSP891C01" to "PCNSS891C01."
14-2	Para 14.1.2.4	Change "PCNSP891C01" to "PCNSS891C01."

14-3	Para 14.2.2.1	Change "PCNSP891D01" to "PCNSS891D01."
14-4	Para 14.2.2.4	Change "PCNSP891D01" to "PCNSS891D01."
20-6	Para 20.1.2.4a, Col 8	Change "|Input SP891-BU1" in JCL samples to "|Input SS891-BU1."
	Para 20.1.2.4b, Col 8	Change "|Input SP891-BU1" to "|Input SS891-BU1."
20-8	Fig 20-01	Change "SP891-BU1" to "SS891-BU1."
		Change "PCNSP891BU1" to "PCNSS891BU1."
22-1	Para 22.1.1	Change "$ TAPE9 FA,X1D(Tape to be certified)" to "$ TAPE9 FA,X1D,,nnnnn(Enter reel nr of tape to be certified)."

BY ORDER OF THE SECRETARY OF THE AIR FORCE

OFFICIAL

LEW ALLEN, JR., General, USAF
Chief of Staff

VAN L. CRAWFORD, JR., Colonel, USAF
Director of Administration

PART ONE - GENERAL INFORMATION

SECTION 1. INTRODUCTION

1.1 Purpose of Users Manual. The objective of this Users Manual for H6000 utility software is to provide the users with the information necessary to effectively use the H6000 programs and subroutines provided.

1.2 Project References. The H6000 utility software consists of standard Air Force programs and subroutines which support H6000 users Air Force-wide. Some of these programs and subroutines were previously documented by the Directorate of Systems Development, AFDSDC, Gunter AFS, Alabama in H6000 User Advisories.

1.3 Terms and Abbreviations:

CCW - Compaction Control Word
SSF - Standard System Format
DCS - Data Compaction System
CC - Card Column(s)
BCD - Binary Coded Decimal
ISP - Indexed Sequential Processing
I/O - Input/Output

1.4 Security and Privacy. The security and privacy classification will be determined by the input and user.

PART TWO - DATA COMPACTION SYSTEM (DCS)

SECTION 3. SYSTEM SUMMARY

3.1 System Application. The purpose of the DCS is to conserve limited data storage space. Operational improvements provided by the DCS include reduced processor time and reduced data storage space resulting in an overall reduction in the cost per job run. The storage space savings can be valuable when disk storage is involved and can enhance disk storage as a viable alternative to other modes of storage. Since I/O time is less on a compacted file, I/O error probability is reduced. The functions of the DCS are the compaction and decompaction of SSF files, and the reading and writing of compacted files by user COBOL programs or programs that use the File and Record Control (GFRC) facility. The subroutines allow compacted file interface with negligible modification to existing programs.

3.2 System Operation. N/A.

3.3 System Configuration. The DCS was written for use on the H6000 computer system with tape and disk I/O devices.

3.4 System Organization. The DCS consists of two stand-alone programs (ZAP1FO and ZAP2FO) and four subroutines (ZAP3FO, ZAP4FO, ZAP5FO, and ZAP6FO).

3.4.1 ZAP1FO - File Compaction Program. ZAP1FO uses SSF files as input to produce compacted SSF files.

3.4.2 ZAP2FO - File Decompaction Program. ZAP2FO uses DCS compacted SSF files as input to produce decompacted SSF files.

3.4.3 ZAP3FO - Write Compacted File Subroutine. User programs call this subroutine to write compacted output files.

3.4.4 ZAP4FO - Read Compacted File Subroutine. User programs call this subroutine to read a compacted file.

3.4.5 ZAP5FO - COBOL Program Write Compacted File Subroutine. User COBOL programs call this subroutine to write compacted output files.

3.4.6 ZAP6FO - COBOL Program Read Compacted File Subroutine. User COBOL programs call this subroutine to read a compacted file.

3.5 Performance. Several files and associated programs were tested and evaluated using the DCS. The following information describes the system performance capabilities.

3.5.1 Test Analysis. The results of the evaluation were variable. In many cases, the savings realized by the use of the DCS were substantial. In other cases, the dollar savings were too small to be of value, but in all cases space was saved. The savings possible are dependent on the sequence of the file. For example, the greatest optimization resulted with system A as shown in para 3.5.2, an Air Force personnel file. The original 5-reel file was compacted and tested with an associated inquiry program, with savings amounting to over 50% of the original dollar cost.

3.5.2 Performance Data. Processor times shown below include normal processing by the program, which in most cases is the largest portion of the time shown.

	Size In Blocks	Savings In Space	Proc Time	Lapse Time	Cost
SYSTEM A					
Original	40372		.3906	1.250	54.78
Compacted	7864	81%	.3434	.250	25.00
SYSTEM B					
Original	14186		.0465	.251	6.94
Compacted	6480	54%	.0463	.091	4.38
SYSTEM C					
Original	2737		.1048	.322	7.97
Compacted	797	74%	.1046	.295	7.89
SYSTEM D					
Original	3869		.1391	.210	7.01
Compacted	2267	42%	.1485	.180	6.61
SYSTEM E					
Original	239		.0067	.026	N/A
Compacted	65	63%	.0034	.021	N/A
SYSTEM F					
Original	557		.0325	.104	N/A
Compacted	265	52%	.0265	.049	N/A

3.5.3 Test Conclusions. In most cases, significant cost savings should be realized. In all cases, space should be saved.

3.6 Data Base. The DCS supports all forms of data and data files that are in SSF, including all files previously compacted by the DCS.

3.7 General Descriptions of Inputs, Processing, Outputs:

3.7.1 Inputs. Any file in SSF which the user desires to have compacted for the purpose of conserving storage space and/or reducing the dollar cost of processing. Inputs may also be compacted files which were created by this system.

3.7.2 Processing. The DCS programs and subroutines provide the capability to build compacted files by the elimination of redundant words of data between successive records of a file (ZAP1FO), to decompact a compacted file (ZAP2FO), and to perform compacted file read/write operations (ZAP3FO, ZAP4FO, ZAP5FO, and ZAP6FO). Decompacted SSF files are output from ZAP2FO, ZAP4FO, and ZAP6FO. No other data manipulation is performed.

3.7.3 Outputs. Compaction records are written as output by ZAP1FO, ZAP3FO, and ZAP5FO, in system standard format as 318 word records (except the last record which may be shorter). Each compaction record contains multiple logical records. The first word of each input record is a compaction control word (CCW) followed by zero to 59 words of new data (words which differ from those in the previous record). The bits of the CCW which are set to one correspond to those word locations which are to contain the new words when the record is expanded. The last bit is used to

indicate continuation when the original records are longer than 35 words. In this case, additional CCWs are used as required to compact the entire record. For example:

RECORD 1 - AABBCCDDEEFF
RECORD 2 - AAABBBDDEEFG
Compacted RECORD 2 - CCABBG

a. After compaction, record 2 may be read as a five-word record where cc is a 36-bit CCW with bits set on corresponding to the word position of the four different words in the record.

b. In the example above, the compaction code has the 3rd, 5th, 6th, and 12th bits on (cc = 001011000001).

c. In the example, a 10-word record is compacted to five words.

d. The record control word of the original system standard format record is considered as part of the data that is compacted. This permits processing of variable length SSF records.

SECTION 4. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

4.1 ZAP1FO - File Compaction Program. The function of this program is to convert SSF files to compacted SSF files.

4.1.1 Initiation Procedures. To compact SSF files, the user may execute the following JCL:

```
Col 1          Col 8          Col 16
    $              IDENT          (USER SPECIFIED)
    $              USERID         (USER SPECIFIED)
    $              LIBRARY        LA
    $              USE            ZAP1FO,ZAP3FO
    $              ENTRY          ZAP1FO
    $              EXECUTE
    $              LIMITS         ,5K
    $              TAPE9          LA,XØD,,nnnnn,,User Library
    $              TAPE9          FA,X1D,,nnnnn,,FZAP1FOAU
    $              TAPE9          F1,X2D,,99999,,FZAP1FO1U
    (OPTIONAL PARAMETER CARD)
    $              ENDJOB
```

4.1.1.1 Program Notes:

a. Since the compacted file is in SSF, it may be processed via standard utilities for copying or dumping. However, its sequence must not be disturbed, or else recovery of the original is impossible.

b. Input is on file FA; output is on file F1.

c. After compaction, the program will display on SYSOUT a count of input and output blocks. This can be used as an indication of how much compaction is realized.

d. Subroutine ZAP3FO is required by ZAP1FO.

e. Library tape LA is in sequential R* format.

f. The user may set the file-ID desired on an output tape by using the following parameter card. Enter the PCN in cc 1-11 and the desired file-ID in cc 19-30. If the parameter card is not included, the file-ID will be *CONDOUT.

```
Col 1          Col 8          Col 16         Col 19
    $              DATA           CA
PCNSP891P11                                      FZAP1FOAUØO
```

4.1.1.2 References. H6000 Control Card Manual #BS19.

4.1.2 Staff Input Requirements. Input is a SSF file. User requirements will determine frequency of processing.

4.1.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.1.2.2 Composition Rules. The input files may be any type of SSF file in which records are 518 words or less in length.

4.1.2.3 Input Vocabulary. N/A.

4.1.2.4 Sample Inputs. N/A.

4.1.3 Output Requirements. Output will be a compacted file in SSF format. User requirements determine the frequency of processing and disposition.

4.1.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.1.3.2 Sample Outputs. N/A.

4.1.3.3 Output Vocabulary. N/A.

4.1.4 Utilization of System Outputs. User determined.

4.1.5 Recovery and Error Correction Procedures. N/A.

4.2 ZAP2FO - File Decompaction Program. The function of this stand-alone program is to convert compacted files to decompacted SSF.

4.2.1 Initiation Procedures. To convert compacted files to decompacted SSF, the user may execute the following JCL.

```
Col 1          Col 8          Col 16
    $              IDENT          (USER SPECIFIED)
    $              USERID         (USER SPECIFIED)
    $              LIBRARY        LA
    $              USE            ZAP2FO,ZAP4FO
    $              ENTRY          ZAP2FO
    $              EXECUTE
    $              LIMITS         ,5K
    $              TAPE9          LA,XØD,,nnnnn,,User Library
    $              TAPE9          FA,X1D,,nnnnn,,FZAP2FOAU
    $              TAPE9          F1,X2D,,99999,,FZAP2FO1U
    (OPTIONAL PARAMETER CARD)
    $              ENDJOB
```

4.2.1.1 Program Notes:

a. Input is on file FA, output is on file F1.

b. Subroutine ZAP4FO is required by ZAP2FO.

c. Library LA is in sequential R* format.

d. The user may set the file-ID desired on an output tape by using the following parameter card. Enter the PCN in cc 1-11 and the desired file-ID in cc 19-30. If the parameter card is not included, the file-ID will be *EXPAND-OUT.

```
Col 1    Col 8          Col 16        Col 19
   $       DATA            CA
   PCNSP891P21                          FZAP2FOAUØO
```

4.2.1.2 References. H6000 Control Card Reference Manual #BS19.

4.2.2 Staff Input Requirements. Input is any file that has been compacted by this DCS. User requirements will determine the frequency of processing.

4.2.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.2.2.2 Composition Rules. N/A.

4.2.2.3 Input Vocabulary. N/A.

4.2.2.4 Sample Inputs. N/A.

4.2.3 Output Requirements. Output is a SSF file expanded from a DCS compacted file. User requirements will determine the frequency of processing and disposition.

4.2.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.2.3.2 Sample Outputs. N/A.

4.2.3.3 Output Vocabulary. N/A.

4.2.4 Utilizatior of System Outputs. User determined.

4.2.5 Recovery and Error Correction Procedures. N/A.

4.3 ZAP3FO - Write Compacted File Subroutine. The function of this subroutine is to permit user GMAP, FORTRAN, and other programs that use the file and record control facility to write compacted files. The subroutine expects the user to write SSF records which will be compacted and put on the output file.

4.3.1 Initiation Procedures. ZAP3FO contains two routines; a file open routine, COTOPN, and a file close routine, COTCLO. To write compacted files, the user must replace his CALL OPEN and CALL CLOSE statements for the files to be compacted with the following:

```
CALL COTOPN (FILEA,FILEB,...,etc.)
CALL COTCLO (FILEA,FILEB,....,etc.)
```

4.3.1.1 Program Notes.

a. FILEA,FILEB,...,etc., must be described as SSF.

b. Limit is 15 files per program.

c. COTOPN saves a current record image in available memory. The size of this record is the maximum record size of the file, plus one word. If sufficient memory is not available, an extra K of memory may be required.

d. The JCL must include the following:

```
Col 1       Col 8          Col 16

    $           LIBRARY        LA
    $           USE            USER PGM,ZAP3FO
    $           ENTRY          USER PGM
    $           EXECUTE
    $           TAPE9          LA,XØD,,nnnnn,,User Library
```

* e. Some H6000 versions of FORTRAN do not use file and record control (GFRC). Data compaction will not work with those versions.

4.3.1.2 References. H6000 Control Card Reference Manual #BS19.

4.3.2 Staff Input Requirements. Input is any type of SSF file. User requirements will determine the frequency of processing.

4.3.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.3.2.2 Composition Rules. N/A.

4.3.2.3 Input Vocabulary. N/A.

4.3.2.4 Sample Inputs. N/A.

4.3.3 Output Requirements. Output will be a SSF file in compacted form. User requirements will determine the frequency of processing and disposition.

4.3.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.3.3.2 Sample Outputs. N/A.

4.3.3.3 Output Vocabulary. N/A.

4.3.4 Utilization of Systems Outputs. User determined.

4.3.5 Recovery and Error Correction Procedures. N/A.

4.4 ZAP4FO - Read Compacted File Subroutine. This subroutine permits user GMAP, FORTRAN, and other programs that use the file and record control facility to read compacted files. Records are decompacted and provided to the user program in system standard format.

4.4.1 Initiation Procedures. ZAP4FO contains one routine, CINOPN. To read compacted files, the user should execute the following:

CALL CINOPN (FILEA,FILEB,...etc.)

4.4.1.1 Program Notes:

a. CINOPN must be called before any of the files in question are opened.

b. The subroutine automatically decompacts and passes a system standard format record to the using program for each 'GET' of the compacted files.

c. FILEA,FILEB,...etc., must be described as system standard format.

d. CINOPN saves a current record image in available memory. The size of this image is the maximum record size of that file plus two words. If sufficient memory is not available, an extra K of memory may be required.

e. JCL must include the following:

```
Col 1       Col 8          Col 16

  $           LIBRARY        LA
  $           USE            USER PGM,ZAP4FO
  $           ENTRY          USER PGM
  $           EXECUTE
  $           TAPE9          LA,XØD,,nnnnn,,User Library
```

* f. Some H6000 versions of FORTRAN do not use file and record control (GFRC). Data compaction will not work with those versions.

4.4.1.2 References. H6000 Control Card Reference Manual #BS19.

4.4.2 Staff Input Requirements. Input is a system standard format file previously compacted by this DCS. User requirements will determine the frequency of processing.

4.4.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.4.2.2 Composition Rules. Limit is 15 files per program.

4.4.2.3 Input Vocabulary. N/A.

4.4.2.4 Sample Inputs. N/A.

4.4.3 Output Requirements. Output is a SSF file expanded from a DCS compacted file. User requirements will determine the frequency of processing and disposition.

4.4.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.4.3.2 Sample Outputs. N/A.

4.4.3.3 Output Vocabulary. N/A.

4.4.4 Utilization of System Outputs. User determined.

4.4.5 Recovery and Error Correction Procedures. N/A.

4.5 ZAP5FO - COBOL Program Write Compacted File Subroutine. This subroutine permits user COBOL programs to write compacted files. The subroutine expects the user to write system standard format records, which will be compacted and written in the output file.

4.5.1 Initiation Procedures. ZAP5FO contains two routines; a file open routine, COMPOT, and a file close routine, COMPCL. To write compacted files, the user must follow the OPEN and precede the CLOSE statements for the files to be compacted with the following:

```
CALL COMPOT USING 01-VARIABLE-A, 01-VARIABLE-B,...etc.
CALL COMPCL USING 01-VARIABLE-A, 01-VARIABLE-B,...etc.
```

4.5.1.1 Program Notes:

a. 01-VARIABLE-A,01-VARIABLE-B,...etc. must be 01-level record descriptors of the output files to be compacted.

b. A process area must be explicitly or implicitly defined.

c. Limit is 15 files per program.

d. COMPOT saves a current record image in available memory. The size of this image is the maximum record size plus one word. If sufficient memory is not available, an extra ᴷ of memory may be required.

e. JCL must include the following:

```
Col 1       Col 8          Col 16

   $            LIBRARY        LA
   $            USE            USER PGM,ZAP5FO
   $            ENTRY          USER PGM
   $            EXECUTE
   $            TAPE9          LA,X#D,,nnnnn,,User Library
```

f. $ LOWLOAD cannot be used.

4.5.1.2 References. H6000 Control Card Reference Manual #BS19.

4.5.2 Staff Input Requirements. Inputs are SSF records generated from a user COBOL program for compaction into a file. User requirements will determine frequency of processing.

4.5.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.5.2.2 Composition Rules. N/A.

4.5.2.3 Input Vocabulary. N/A.

4.5.2.4 Sample Inputs. N/A.

4.5.3 Output Requirements. Output is a compacted SSF file. User requirements will determine the frequency of processing and disposition.

4.5.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.5.3.2 Sample Outputs. N/A.

4.5 3.3 Output Vocabulary. N/A.

4.5.4 Utilization of System Outputs. User Determined.

4.5.5 Recovery and Error Correction Procedures. N/A.

4.6 ZAP6FO - COBOL Program Read Compacted File Subroutine. This subroutine permits user COBOL programs to read compacted files. Records are decompacted and provided to the user program in system standard format.

4.6.1 Initiation Procedures. ZAP6FO contains one routine COMPIN. To read compacted files, the user should follow the OPEN statement with the following:

CALL COMPIN USING 01-VARIABLE-AL, 01-VARIABLE-B.,,,etc.

4.6.1.1 Program Notes:

a. 01-VARIABLE-A,01-VARIABLE-B.,,,etc., must be an 01-level record descriptor of the input files which are to be read (reference ZAP5FO example).

b. A process area must be explicitly or implicitly defined.

c. Files must be closed with the normal COBOL close statement.

d. JCL must include the following:

Col 1	Col 8	Col 16
$	LIBRARY	LA
$	USE	USER PGM,ZAP6FO
$	ENTRY	USER PGM
$	EXECUTE	
$	TAPE9	LA,XØD,,nnnnn,,User Library

4.6.1.2 References. H6000 Control Card Reference Manual #BS19.

4.6.2 Staff Input Requirements. Input is a SSF file previously compacted by this DCS. User requirements will determine the frequency of processing.

4.6.2.1 Input Formats. Reference H6000 File and Record Control Manual #DD07.

4.6.2.2 Composition Rules. Limit is 15 files per program.

4.6.2.3 Input Vocabulary. N/A.

4.6.2.4 Sample Inputs. N/A.

4.6.3 Output Requirements. Output is a SSF file expanded from a DCS compacted file. User requirements will determine the frequency of processing and disposition.

4.6.3.1 Output Formats. Reference H6000 File and Record Control Manual #DD07.

4.6.3.2 Sample Outputs. N/A.

4.6.3.3 Output Vocabulary. N/A.

4.6.4 Utilization of System Outputs. User determined.

4.6.5 Recovery and Error Correction Procedures. N/A.

PART FOUR - CARD UTILITIES

SECTION 7. SYSTEM SUMMARY

7.1 System Application. Many MAJCOM standards and unique system are required to process BCD card decks for/from non-WWMCCS ADPE. Since the H6000 WWMCCS card reader and punch recognize a 64 character ASCII subset, a software interface is required to enable processing of BCD cards containing punches outside the WWMCCS 64 characters ASCII subset. The most common punches are the (12) (plus Ø) and (11) (minus Ø). (Ø) (Ø)

7.2 System Operation. N/A.

7.3 System Configuration. These utility programs were written for use on H6000 WWMCCS ADPE.

7.4 System Organization. The system consists of two standard programs.

7.4.1 ZAAØFO - H6000 Binary to BCD Card Input. This program reads a BCD card deck in binary mode, transliterates the data and writes the card images to a file in SSF.

7.4.2 ZABØFO - H6000 BCD to Binary Card Punch. This program punches a binary card deck from SSF file.

7.5 Performance. N/A.

7.6 Data Base. N/A.

7.7 General Description of Inputs, Processing, Outputs:

7.7.1 Inputs:

7.7.1.1 ZAAØFO:

a. PCN Parameter Cards.

b. TAKE Parameter Card. This card allows the user to substitute invalid input punches with a specified value.

c. Value of ID Parameter Card. This card allows the user to specify a value of ID for output tape files.

d. Replacement Parameter Card. This card(s) allows the user to modify standard input character transliteration.

e. User Card file in BCD format.

7.7.1.2 ZABØFO:

a. PCN Parameter Card.

b. Replacement Parameter Card. This card allows the user to modify standard output character transliterations.

c. Users tape or disk file in Standard System Format.

7.7.2 Processing:

7.7.2.1 ZAAØFO. BCD card file is read, transliterated, and written to a tape or disk file in Standard System Format.

7.7.2.2 ZABØFO. User tape or disk file in Standard System Format is read, transliterated and a binary card deck is produced.

7.7.3 Outputs:

7.7.3.1 ZAAØFO. User tape or disk file in Standard System Format.

7.7.3.2 ZABØFO. User card file is BCD format.

SECTION 8. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

8.1 ZAAØFO - H6000 Binary to BCD Card Input. This program reads BCD card decks from non-H6000 ADPE that contains punches that are outside the WWMCCS 64-character ASCII subset.

8.1.1 Initiation Procedures. To initiate the execution of ZAAØFO, the user may prepare and execute the Job Control Language (JCL) as outlined in para 8.1.2.4.

8.1.1.1 Program Notes:

a. ZAAØFO reads BCD cards in binary mode, via a dedicated card reader, transliterates the data and writes the card images to a user defined tape or disk file in SSF. Invalid characters detected on input are replaced by the character "Ø" and a message is displayed at SYSOUT indicating the location of the invalid character.

b. An option is provided to allow the user to override the standard transliteration table. This option is activated by setting SWITCH 1 and utilizing the Replacement Parameter Card.

c. An option is provided to allow the user to replace invalid input characters with other than the default of "Ø". This option is activated by setting SWITCH 2 and utilizing the TAKE Parameter Card.

d. An option is provided to allow the user to place a Value of ID in the internal label of the output tape file. This option is activated by setting SWITCH 3 and utilizing the Value of ID Parameter Card.

8.1.1.2 References:

a. H6000 Control Cards, BS19.

b. H6000 File System, BR38.

c. H6000 GCOS, BR43.

8.1.2 Staff Input Requirements:

8.1.2.1 Input Formats:

a. PCN Parameter Card. (Ref Figure 8-01).

b. Value of ID Parameter Card. (Ref Figure 8-02).

c. PCN Parameter Card. Required (Ref Figure 8-03).

d. TAKE Parameter Card. (Ref Figure 8-04).

e. Replacement Parameter Card. (Ref Figure 8-05). This Parameter Card contains a series of definition fields, each three characters long. Each card may contain up to 26 definition fields.

(1) To alter the standard transliteration table to define a replacement character, the first column contains the punch, the second and third columns contain the internal OCTAL value that the punch is to be converted to.

Col 1-3

(12) (12)
(Ø) 37 - This example parameter card would change the table so that a (Ø) punch would be converted to an internal value 37 OCTAL.

(2) To delete an octal value from the table (making it illegal); the first column contains the punch, the second and third columns are blank.

Col 1-3

(12)
(Ø) ⊭⊭

(3) To replace a punch with another punch leaving the OCTAL conversion value unchanged; the first column contains the old punch, the second column contains the new punch, and the third column is blank.

Col 1-3

(12) (12)
(6) (Ø)⊭ A (12) punch will be converted to an OCTAL 6Ø and
(8) (1)

(12)
A (6) punch will be considered illegal.
(8)

8.1.2.2 Composition Rules. N/A.

8.1.2.3 Input Vocabulary. N/A.

8.1.2.4 Sample Inputs:

a. A sample job stream to read a BCD card deck and produce a Standard System Format tape file with a value ID of "FMJØ1FO3X":

```
Col 1       Col 8          Col 16
$           IDENT          (USER SPECIFIED)
$           USERID         (USER SPECIFIED)
$           LIBRARY        LB
$           USE            ZAAØFO
$           ENTRY          ZAAØFO
$           EXECUTE        ON3
$           LIMITS         ,5K
$           PRMFL          LB,R,S,(USER-SPECO
$           READ           C1
$           TAPE9          F1,X1D,,99999,,FMJØ1FO3X
PCNSP891AØØ
FMJØ1FO3X                  ID
$           ENDJOB
```

b. A sample job stream to read a BCD deck, alter the translation table, and to use an (11) punch (*) as a replacement value for any invalid characters found in
(4)
(8)
the data.

```
Col 1          Col 8              Col 16
    $              IDENT              (USER SPECIFIED)
    $              USERID             (USER SPECIFIED)
    $              LIBRARY            LB
    $              USE                ZAAØFO
    $              ENTRY              ZAAØFO
    $              EXECUTE            ON1,ON2
    $              LIMITS             ,SK
    $              PRMFL              LB,R,S,(USER-SPEC)
    $              READ               C1
    $              PRMFL              F1,R/W,S,(USER-SPEC)
    $              ENDJOB
PCNSP891ØØ1
TAKE=*
(REPLACEMENT PARAMETER CARD)
***EOF
```

The TAKE (PARAMETER CARD) is input as a separate card deck when ZAAØFO dedicates the card reader.

Col 1-4 of the Replacement Parameter Card.

$\binom{12}{\emptyset}$ 37 $\binom{11}{\emptyset}$ parameter card to convert $\binom{12}{\emptyset}$ to

OCTAL 37 and to delete $\binom{11}{\emptyset}$ making it illegal.

8.1.3 Output Requirements:

8.1.3.1 Output Formats. N/A.

8.1.3.2 Sample Outputs. N/A.

8.1.3.3 Output Vocabulary. N/A.

8.1.4 Utilization of System Outputs. N/A.

8.1.5 Recovery and Error Correction Procedures. N/A.

8.2 ZABØFO - H6000 BCD to Binary Card Punch. This program reads a SSF 14 word file, expands it to 27 word binary records performing specified character transliterations and punches these converted records in binary mode.

8.2.1 Initiation Procedures. To initiate the execution of ZABØFO, the user may prepare and execute the Job Control Language (JCL) as outlined in para 8.2.2.4.

8.2.1.1 Program Notes. The function of this program is to output BCD card decks containing nonstandard ASCII characters. This table is initialized for the standard ASCII character set with the exception of OCTAL 6Ø and 4Ø. OCTAL 6Ø will punch as a $\binom{12}{\emptyset}$ and 4Ø as an $\binom{11}{\emptyset}$.

The user can modify this table to punch different characters for specified OCTAL values. The same parameter card format and rules described for ZAAØFO apply to ZABØFO; however, only the replace (ref paragraph 8.1.2.1.4) capability is provided. The definition field in this case has a different meaning; the OCTAL value specified in the second and third positions relates to the internal OCTAL value and the first position specifies the character to be punched for that internal OCTAL value. If parameter cards are used, one of the following $ Control Cards must be included in the job stream.

```
Col 1          Col 8              Col 16
    $              READ               CA
```

This will dedicate the card reader (since they must be read in binary mode). Or

```
Col 1        Col 8          Col 16
  $            PRMFL          FA,R/W,S,(USER-SPEC)
```

8.2.1.2 References:

a. H6000 Control Cards, BS19.

b. H6000 File System, BR38.

c. H6000 GCOS, BR43.

8.2.2 Staff Input Requirements:

8.2.2.1 Input Formats:

a. PCN Parameter Card. (Ref Figure 8-06).

b. Replacement Parameter Card. (Ref Figure 8-07).

8.2.2.2 Composition Rules. N/A.

8.2.2.3 Input Vocabulary. N/A.

8.2.2.4 Sample Inputs:

a. A sample job stream to produce a BCD card deck for a standard system format input tape file:

```
Col 1        Col 8          Col 16
  $            IDENT          (USER SPECIFIED)
  $            USERID         (USER SPECIFIED)
  $            LIBRARY        LB
  $            USE            ZABØFO
  $            ENTRY          ZABØFO
  $            EXECUTE
  $            LIMITS         ,5K
  $            TAPE9          LB,XØD,,nnnnn,,USER-LIBRARY
  $            TAPE9          F1
  $            READ           CA
  $            PUNCH          C1 (OR $ SYSOUT C1)
  $            ENDJOB
 (Parameter Card if used is input as a separate card deck when ZABØFO
  dedicates the card reader)
  $            ENDJOB
```

Col 1-9 of Replacement Parameter Card:

(12) (12) (12)
(6) 6Ø (Ø) 37 (11) 40 Parameter card to punch (6) for
(8) (8)

OCTAL 6Ø instead of (12) (Ø), punch a

(12) (Ø) for OCTAL 37 and punch (11)

for an OCTAL 4Ø instead of (11) (Ø)

punch.

b. The same job using a disk file as input instead of a tape file.

```
Col 1          Col 8            Col 16
    $              IDENT            (USER SPECIFIED)
    $              USERID           (USER SPECIFIED)
    $              LIBRARY          LB
    $              USE              ZABØFO
    $              ENTRY            ZABØFO
    $              EXECUTE
    $              LIMITS           ,5K
    $              TAPE9            LB,XØL,,nnnnn,,USER-LIBRARY
    $              PRMFL            F1,R,S,(USER-SPEC)
    $              PUNCH            C1(OR $ SYSOUT C1)
    $              READ             CA(OR $ PRMFL      FA,R/W,S,(USER-SPEC))
    $              ENDJOB
    (Parameter Card if used is input as a separate card deck when ZABØFO
    dedicates the card reader.)
    $              ENDJOB
```

Col 1-9 of Replacement Parameter Card:

```
(12)    (12)
( 6) 6Ø ( Ø) 37 (11) 4Ø                    (ref paragraph 8.2.2.4.a)
( 8)
```

c. Parameter cards may be put to a perm-file for ease of execution. A sample job stream to put parameter cards to a perm-file.

```
Col 1          Col 8            Col 16
    $              CONVER
    $              READ             IN        (See Note 1)
    $              INPUT            MBIN
    $              PRMFL            OT,R/W,S,(USER-SPEC)
    $              ENDJOB
    ***EOF
```

NOTE 1: The $ READ control card dedicates the card reader. After the card reader is assigned to the CONVER activity, load card reader with parameter card(s) and a EOF control card and press operate.

8.2.3 Output Requirements:

8.2.3.1 Output Formats. N/A.

8.2.3.2 Sample Outputs. N/A.

8.2.3.3 Output Vocabulary. N/A.

8.2.4 Utilization of System Outputs. N/A.

8.2.5 Recovery and Error Correction Procedures. N/A.

RECORD LAYOUT

FILE TITLE	RECORD TITLE		CLASSIFICATION
File Identification CZAAØFOAU	PCN Parameter Card		Unclassified
			RCS/PCN SP891-AØØ
FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
PIC	01 - 11	AN	Enter "PCNSP891AØØ"
Filler	12 - 80		Blank

Note: When the Value of ID card is used, the PCN card is required as the first of the deck.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-01. PCN Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE		CLASSIFICATION
File Identification CZAAØFOAU	VALUE OF ID Parameter Card		Unclassified
			RCS/PCN SP891-AØØ
FILE DESCRIPTION	**RECORD POSITIONS**	**TYPE/CLASS**	**SPECIAL INSTRUCTIONS**
FILE-ID	Ø1 - 12	AN	Enter FILE-ID.
Filler	13 - 16		Blank.
Card Identity	17 - 18	AL	Enter "ID".
Filler	19 - 8Ø		Blank.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-02. VALUE OF ID Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE	CLASSIFICATION
Parameter Card/ Binary Deck CZAAØFOBU	PCN Parameter Card	Unclassified
		RCS/PCN SP891-AØ1

FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
PIC	Ø1 - 11	AN	Enter "PCNSP891AØ1".
Filler	12 - 8Ø		Blank.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-03. PCN Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE		CLASSIFICATION
Parameter Card/ Binary Deck CZAAØFOBU	TAKE Parameter Card		Unclassified
			RCS/PCN SP891-AØ1
FILE DESCRIPTION	**RECORD POSITIONS**	**TYPE/CLASS**	**SPECIAL INSTRUCTIONS**
Card Identity	Ø1 - Ø5	AN	Enter "TAKE=".
Replacement character	Ø6	AN	Any valid character to be used as a replacement value for invalid characters.
Filler	Ø7 - 8Ø		Blank.
Note: The character (!) octal 77 should not be used as a replacement character, as it could be read as a printer slew character.			

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-04. TAKE Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE	CLASSIFICATION
Parameter Card/ Binary Deck CZAAØFOBU	Replacement Parameter Card	Unclassified
		RCS/PCN SP891-AØ1

FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
	ALTER OPTION		
Definition Field #1 Input Character	Ø1	AN	Character to be modified.
Internal OCTAL Value	Ø2 - Ø3	UN	OCTAL value that input character punch is to be converted to.
Definition Field #2 - #26	Ø4 - 78	AN	Same format as Definition Field #1, if unused Blank.
Filler	79 - 8Ø		Blank.
	DELETE OPTION		
Definition Field #1 Input Character	Ø1	AN	Character to be deleted.
Filler	Ø2 - Ø3		Blank.
Definition Fields #2 - #26	Ø4 - 78	AN	Same format as Definition Field #1, if unused Blank.
Filler	79 - 8Ø		Blank.
	REPLACE OPTION		
Definition Field #1 Input Character	Ø1	AN	Character to be replaced.
Output Character	Ø2	AN	New character replacing input character.
Filler	Ø3		Blank.
Definition Fields #2 - #26	Ø4 - 78	AN	Same format as Definition Field #1, if unused Blank.
Filler	79 - 8Ø		Blank.

AF FORM AUG 70 1190 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-05. Replacement Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE	CLASSIFICATION
Parameter Card CZABØFOAU	PCN Parameter Card	Unclassified
		RCS/PCN SP891-BØ1

FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
PIC	Ø1 - 11	AN	Enter "PCNSP891BØ1".
Filler	12 - 8Ø		Blank.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-06. PCN Parameter Card

RECORD LAYOUT

FILE TITLE	RECORD TITLE		CLASSIFICATION
Parameter Card CZABØFOAU	Replacement Parameter Card		Unclassified
			RCS/PCN SP891-BØ1
FILE DESCRIPTION	**RECORD POSITIONS**	**TYPE/CLASS**	**SPECIAL INSTRUCTIONS**
Definition Field #1			
Output Character	Ø1	AN	Output character to be punched.
Internal OCTAL Value	Ø2 - Ø3	UN	The OCTAL value of input to be transliterated to the output character.
Definition Fields #2 - #26	Ø4 - 78	AN	Same format as Definition Field #1, if unused Blank.
Filler	79 - 8Ø		Blank.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 8-07. Replacement Parameter Card

PART FIVE - GENERAL PURPOSE TAPE FILE INPUT UTILITY

SECTION 9. SYSTEM SUMMARY

9.1 System Application. The function of the General Purpose Tape File Input Utility (GPTFIU) is to read 7 or 9-track H6000 Standard System or Non-Standard System Format Tape Files and convert them into the opposite format. Character transliterations are performed as specified by the user.

9.2 System Operation. N/A.

9.3 System Configuration. The GPTFIU was written for use on the H6000 computer system with Tape I/O devices.

9.4 System Organization. The GPTFIU consists of two stand-alone programs (ZAT1FO and ZAT2FO).

9.5 Performance. N/A.

9.6 Data Base. N/A.

9.7 General Description of Inputs, Processing, Outputs:

9.7.1 Inputs:

a. ZAT1FO - any 7 or 9-track tape with a logical record size less than 1909 characters, whose records, if variable in length, are unblocked with no control information included.

b. ZAT2FO - any SSF tape or disk file.

9.7.2 Processing. The GPTFIU provides the capability to convert an SSF tape or disk to a Non-Standard System Format Tape and vice-versa.

9.7.3 Outputs:

a. ZAT1FO - any SSF tape or disk file.

b. ZAT2FO - the output file can be written to a 7 or 9-track tape in a variety of modes and densities. True physical record length is not always reflected on the output tape due to H6000 hardware limitations that require all physical record lengths be written modulo 4 (ASCII, EBCDIC) or modulo 6 (Binary, BCD). This limitation requires the padding of physical records with zeros to the next greater word boundary.

(3) Blank Follows - 4

75 %

SECTION 10. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

10.1 ZAT1FO - General Purpose Tape File Input Program. The function of this program is to read a non-H6000/H635 tape, perform specified unblocking and character transliteration and ultimately create a Standard System Format (SSF) tape or disk file.

10.1.1 Initiation Procedures. To initiate the execution of ZAT1FO, the user must introduce the following JCL for execution:

```
Col 1       8         16
    $       IDENT     (USER SPECIFIED)
    $       USERID    (USER SPECIFIED)
    $       LOWLOAD
    $       LIBRARY LA
    $       USE       ZAT1FO
    $       ENTRY     ZAT1FO
    $       EXECUTE
    $       LIMITS    ++,kk
    $       TAPEn     LA,L1D,,nnnnn,,(USER-LIBRARY)
    (OPTIONAL PARAMETER CARDS)
    $       TAPEn     F1,F1D,,nnnnn,,(USER NON-SSF INPUT)
    $       TAPEn     F2,F2D,,,,(USER SSF OUTPUT)
    $       ENDJOB
```

where ++ = the user's estimated run time.
where kk = the user's estimated core requirements.

Program ZAT1FO operates in 6K with 1920 character/1280 byte input buffers. Add 1K for each additional 3000 character/2000 bytes contained in the input physical record.

10.1.1.1 Program Notes. N/A.

10.1.1.2 References:

a. H6000 Control Cards Reference Manual, DD31.

b. H6000 File and Record Control Manual, DD07.

10.1.2 Staff Input Requirements. User requirements will determine the frequency of processing.

10.1.2.1 Input Formats. If the optional parameter cards are not provided, the following default characteristics are assumed for the input tape:

Unlabeled
Binary/odd parity
Low density
Unblocked 80-character records
H200 character set transliteration
No error recovery
Single file (up to 99 reels input)
No pad character deletion

a. PIC Card. If parameter cards are input, the PIC card must be the first card (Figure 10-01).

b. Input/Output Card. This parameter card must be the second card and is used to define the output file value of ID (if mag tape) and all the physical characteristics of the input tape file (Figure 10-02).

(5)

75%

c. Transliteration Card(s). This card(s) is optional. If other than standard transliteration table values are to be utilized, one or more of these cards must be provided immediately after the Input/Output Card. Ref Figure 10-05. Also ref Figures 10-07 thru 10-09 for standard transliteration table values.

10.1.2.2 Composition Rules. The maximum input tape physical record size is 24,570 character/16,380 bytes. The maximum logical record size is 1908 characters/bytes.

10.1.2.3 Input Vocabulary. N/A.

10.1.2.4 Sample Inputs. N/A.

10.1.3 Output Requirements. User requirements will determine the disposition of the output file(s).

10.1.3.1 Output Formats. Output will be a SSF tape or disk file. Ref DD07 for formats.

* 10.1.3.2 Sample Outputs. A parameter card listing is produced when parameters are input. All parameter cards will be edited to replace the H6000 slew character "!" (77_8) with "\" (37_8) prior to printing (Figure 10-11).

10.1.3.3 Output Vocabulary. N/A.

10.1.4 Utilization of System Outputs. N/A.

10.1.5 Recovery and Error Correction Procedures. N/A.

10.2 ZAT2FO - General Purpose Tape File Output Program. The function of this program is to read an H6000/H635 SSF tape or disk file, perform specified labeling, blocking and character transliteration, and ultimately create a non-H6000/H635 tape file.

10.2.1 Initiation Procedures. To initiate the execution of ZAT2FO, the user must introduce the following JCL for execution:

```
Col 1       8       16
    $       IDENT   (USER SPECIFIED)
    $       USERID  (USER SPECIFIED)
    $       LOWLOAD
    $       LIBRARY LA
    $       USE     ZAT2FO
    $       ENTRY   ZAT2FO
    $       EXECUTE
    $       LIMITS  ++,kk
    $       TAPEn   LA,L1D,,nnnnn,,(USER-LIBRARY)
    (OPTIONAL PARAMETER CARDS)
    $       TAPEn   F1,F1D,,nnnnn,,(USER SSF INPUT)
    $       TAPEn   F2,F2D,,,,(USER NON-SSF OUTPUT)
    $       ENDJOB
```

where ++ = the user's estimated run time.
where kk = the user's estimated core requirements.

Program ZAT2FO operates in 6K with 1920 character/1280 byte output buffers. Add 1K for each additional 3000 characters/2000 bytes to be contained in the output physical record.

(6)

75%

10.2.1.1 Program Notes. N/A.

10.2.1.2 References:

a. H6000 Control Card Reference Manual, DD31.

b. H6000 File and Record Control Manual, DD07.

10.2.2 Staff Input Requirements. User requirements will determine the frequency of processing.

10.2.2.1 Input Formats. If the optional parameter cards are not provided, the following default characteristics are assumed for the output tape:

Unlabeled
Binary/odd parity
Low density
Unblocked 80-character records
H200 character set transliteration
Single file

a. PIC Card. If parameter cards are input, the PIC card must be the first card (Figure 10-03).

b. Input/Output Card. This parameter card must be the second card and is used to define the input file value of ID (mag tape only) and all the physical characteristics of the output tape (Figure 10-04).

c. Transliteration Card(s). These card(s) are optional. If other than standard transliteration table values are to be utilized, one or more of these cards must be provided immediately following the Input/Output Card (Figure 10-05). Also reference Figure 10-07 thru 10-09 for standard table values.

d. Label Card(s). These card(s) are optional. If labels are specified by the OUTPUT LABELS field (ref Figure 10-04), this card(s) must follow all other cards in the input deck (Figure 10-06).

10.2.2.2 Composition Rules. The SSF input file cannot contain partitioned records.

10.2.2.3 Input Vocabulary. N/A.

10.2.2.4 Sample Inputs. N/A.

10.2.3 Output Requirements. User requirements will determine the disposition of the output tape(s).

10.2.3.1 Output Formats. Output will be a non-H6000/H635 tape file created according to user supplied or default parameters.

10.2.3.2 Sample Outputs. A parameter card listing is produced when parameters are input. All parameter cards will be edited to replace the H6000 slew character "!" (77_8) with "↖" (37_8) prior to printing (Figure 10-12).

10.2.3.3 Output Vocabulary. N/A.

10.2.4 Utilization of System Outputs. N/A.

10.2.5 Recovery and Error Correction Procedures. N/A.

ZAT1FO PIC CARD

FIELD (Col)	DESCRIPTION
PCN(01-11)	Enter "PCNSP891T11"
BLANK(12-80)	

FIGURE 10-01. ZAT1FO PIC Card

(8)

75%

ZATIFO INPUT/OUTPUT CARD

FIELD (Col)	DESCRIPTION
OUTPUT ID (01-12)	If an internal Value of ID is desired (for magnetic tape only) enter the value left justified.
INPUT LABELS (13-18)	The following describes the internal labels contained on the input tape. "NLABEL" or blank - unlabeled. * "SLABEL" - standard H6000 labels. * "ULABEL" - non-standard H6000 label with a file mark following (label is ignored). * "ULABEX" - non-standard H6000 label without a file mark following (label is ignored). * "ULAHTL" - non-standard H6000 header and trailer labels with a file mark following the header and preceding the trailer (labels are ignored). * "ULAHTX" - non-standard H6000 header and trailer labels with no file mark following the header (labels are ignored). * Note 1: The above parameters stating non-standard H6000 header and/or trailer labels implies that the stranger tape input contains labels that were created by a foreign system; i.e., B3500, IBM 360, etc. Note 2: The following modifications may be made to the INPUT LABELS field to allow printing of the header and/or trailer labels on the execution report. No character transliteration will be accomplished but the labels will be edited to replace the H6000 slew character (77$_8$) with (37$_8$). To print header label(s) substitute "PH" for "LA" in col 14-15. To print trailer label(s) substitute "PT" for "HT" in col 16-17.
INPUT MODE/PARITY (19-22)	The following describes the mode/parity of the input tape: "BNRY" or blank - odd parity "ASA9" - 9-track mode. This should be used only for 9-track tape files that are recorded in 8-bit format (EBCDIC, ASCII, etc.) "MBCD" - even parity. Even parity on 7-track input results in a hardware character transliteration (Figure 10-10).
BLANK (23-24)	
INPUT DENSITY (25-28)	The following describes the recording density of input tape: "LOW" or blank - site specified low density (normally 556 bpi) "HIGH" - site specified high density (normally 800 bpi)
BLANK (29-30)	

FIGURE 10-02. ZATIFO Input/Output Card

FIELD (Col)	DESCRIPTION
INPUT BLOCKING FACTOR (31-36)	The following describes the blocking factor of the input tape: "UNBLOK", "VARY␣␣" or blank - a blocking factor of 1 is assumed. If this option is specified, the INPUT RECORD LENGTH field is interpreted as the maximum physical record size and cannot exceed 1908 characters/bytes. "BLKnnn" - where "nnn" is a right-justified numeric integer describing the number of fixed length logical records contained in a physical record.
INPUT RECORD LENGTH (37-42)	The following describes the logical record length on the input tape: Blank - a logical record length of 80 characters/bytes is assumed. "XLnnnn" - where "nnnn" is a right-justified numeric integer. This option is used to specify that each logical record contains "nnnn" characters/bytes for fixed length records or "nnnn" maximum characters for unblocked records.
* Note: When the INPUT PADDING DELETION field is used, the numeric integer of padding deletion must be subtracted from the INPUT RECORD LENGTH field.	
INPUT TRANSLITERATION (43-48)	The following field describes the various transliteration options that may be performed on the input tape: "NONE␣␣" - no input transliteration is to be accomplished
Standard Tables	"H200␣␣" or blank - for use with H200 7-track odd parity tape. "EBCDIC" - for use with B3500/B6700 9-track tape. Can be used for IBM 360 also. "ASCII␣" - for use with H6000 9-track tape recorded in the 64 character UASCI set. "BCL␣␣" - for use with B3500/B6700 7-track tape. Can be used for IBM 360 also.
Standard Table Modification or User Table Replacement	"TRANS6" - 6-bit user supplied table. "TRANS9" - 8-bit user supplied table or standard table modification.
Note: Ref Figure 10-05 for a complete description of the various options available.	
INPUT PARITY ERROR (49-54)	The following field describes the procedures to be accomplished in the event of an uncorrectable input tape error: Blank - will cause an error message to be displayed on the execution report and processing terminated when an uncorrectable error is encountered on the input tape. "USE␣␣␣" - this option permits processing of the physical record in error. "IGNORE" - this option permits deletion of the physical block in error and continuation of processing.
BLANK (55)	

FIGURE 10-02. ZAT1FO Input/Output Card (cont)

FIELD (Col)	DESCRIPTION
INPUT PARITY COUNT (56-57)	The following field is used in conjunction with the INPUT PARITY ERROR field to limit the number of input errors to process: Blank - if the INPUT PARITY ERROR field specifies "USE" or "IGNORE", that option will be applied to the entire file. "nn" - where "nn" is a numeric integer to limit the number of input errors to USE or IGNORE.
INPUT FILE/REEL COUNT (58-60)	The following field description describes the input as a multi-reel file or a multi-file reel: Blank - up to 99 reels of a single file input is assumed. Operator option allows termination after each reel of input. "Mnn" - where "nn" is a numeric integer describing the number of files contained on one input tape reel. "Rnn" or "Vnn" - where "nn" is a numeric integer describing the number of reels contained in the input file.
BLANK (61-66)	
INPUT PADDING DELETION (67)	The following field/subfields description indicate that the input file contains unwanted characters/bytes in the record: " " - no padding. "P" - indicates that one or more of the pad subfield(s) contain a numeric value.

Note: The following pad subfields are programmatically ADDED to the INPUT RECORD LENGTH field when computing the physical record buffer sizes; therefore, the INPUT RECORD LENGTH field must be REDUCED by the numeric integer used by the pad subfields. (INPUT RECORD LENGTH FIELD + PAD SUBFIELD = TOTAL INPUT RECORD LENGTH)

PRE-BLOCK PAD (68-69)	"nn" - where "nn" is a numeric integer describing the number of characters/bytes to be deleted from the beginning of each physical record.
BLANK (70)	
PRE-RECORD PAD (71-72)	"nn" - where "nn" is a numeric integer describing the number of characters/bytes to be deleted from the beginning of each logical record.
BLANK (73)	
POST-RECORD PAD (74-75)	"nn" - where "nn" is a numeric integer describing the number of character/bytes to be deleted from the ending of each logical record.

Note: This subfield should be used with caution when the input file contains variable length records.

FIGURE 10-02. SATIFO Input/Output Card (cont)

ZAT2FO PIC CARD

FIELD (Col)	DESCRIPTION
PCN (01-11)	Enter "PCNSP891T21"
BLANK (12-80)	

FIGURE 10-03. ZAT2FO PIC Card

ZAT2FO INPUT/OUTPUT CARD

FIELD (Col)	DESCRIPTION
INPUT ID (01-12)	If an internal Value of ID check (mag tape only) of the input is desired, enter the Value of ID (left justified).
OUTPUT LABELS (13-18)	The following describes the internal labels that will be created for the output tape: "NLABEL" or blank - unlabeled "ULABEL" - header label(s) only with file mark after header and data. "ULABEX" - header label(s) only with file mark after data. "ULAHTL" - header and trailer label(s) with file marks following header and preceding trailer. "ULAHTX" - header and trailer label(s) with no file mark following header: file mark preceding trailer. "SLABEL" - standard H6000 labels.
Note: If user labels are specified, label parameter cards must be provided (Figure 10-06).	
OUTPUT MODE/ PARITY (19-22)	The following describes the mode/parity of the output tape: "BNRY" or blank - odd parity "ASA9" - 9-track mode. This should be used for creating 9-track files in 8-bit format (EBCDIC, ASCII, etc.) "MBCD" - even parity. Even parity on 7-track output results in a hardware character transliteration (Figure 10-10).
BLANK (23-24)	
OUTPUT DENSITY (25-28)	The following describes the recording density of the output tape: "LOW␣" or blank - site specified low density (normally 556 bpi). "HIGH" - site specified high density (normally 800 bpi).
BLANK (29-30)	
OUTPUT BLOCKING FACTOR (31-36)	The following describes the blocking factor of the output tape: "UNBLOK" or blank - the size of the physical record to be written will be taken from the Record Control Word (RCW) of the input file. When this option is used, the OUTPUT RECORD LENGTH field is ignored. "BLKnnn" - where "nnn" is a right justified numeric integer describing the number of fixed length logical records to be contained in a physical record.

FIGURE 10-04. ZAT2FO Input/Output Card

FIELD (Col)	DESCRIPTION
OUTPUT RECORD LENGTH (37-42)	The following describes the logical record length on the output tape: Blank - a logical record length of 80 characters/bytes is assumed. "XLnnnn" - where "nnnn" is a right justified numeric integer. This option specifies that each logical record contains "nnnn" characters/bytes for fixed length records.
OUTPUT TRANSLITERATION (43-48)	The following field describes the various transliteration option that may be performed on the output tape: "NONE␣␣" - no output transliteration is to be accomplished.
Standard Tables	"H200␣␣" or blank - for use with H200 7-track, odd parity tape. "EBCDIC" - for use with B3500/B6700 9-track tape. Can be used for IBM 360 also. "ASCII␣" - for use with H6000 9-track tape recorded in the 64 character UASCI set. "BCL␣␣␣" - for use with B3500/B6700 7-track tape. Can be used for IBM 360 also.
Standard Table Modification or User Table Replacement	"TRANS6" - a 6-bit user supplied table. "TRANS9" - 8-bit user specified table or standard table modification.

Note: Ref Figure 10-05 for a complete description of the various options available.

BLANK (49-53)

INPUT/OUTPUT MULTI-FILE INDICATOR (54)	"␣" - single file "M" - if the input/output is multi-file

Note: This restricts the SSF input to magnetic tape and requires that both input and output be contained on one physical reel.

BLANK (55)

TAPE UNIT (56) "␣" if MTH 400 model tape units are use.
"1" if MTC 500 or 600 model tape units are used.

BLANK (57-80)

FIGURE 10-04. ZAT2FO Input/Output Card (cont)

TRANSLITERATION CARD

1. "TRANS6" in the TRANSLITERATION FIELD of the INPUT/OUTPUT Card indicates that only one transliteration card is present for the purpose of providing a complete 6-bit transliteration table.

FIELD (Col)	DESCRIPTION
USER SPECIFIED (01-64)	Where Col 1-64 corresponds to the ascending octal values 00-77. Col 1 would contain the H6000 character to replace the input octal value of 00; Col 48 would contain the H6000 character to replace the input octal value of 60. This card represents an entire table; therefore, all 64 columns must be used or erroneous results will occur.

2. "TRANS9" in the TRANSLITERATION FIELD of the INPUT/OUTPUT Card indicates that one or more transliteration cards are present for the purpose of modifying a standard table or providing a complete 8-bit transliteration table.

a. To modify a standard table:

FIELD (Col)	DESCRIPTION
STANDARD TABLE NAME (01-06)	Enter the name of the standard table to be modified. "H200␣␣" "EBCDIC" "ASCII␣" "BCL␣␣␣"
BLANK (07)	
MODIFIERS (08-74)	Enter the octal value of the input character followed by the octal value of the output character desired. All fields are separated by commas with no intervening blanks. To continue on subsequent cards, terminate the first card with a comma (prior to Col 74) and continue on a subsequent card beginning in Col 1.

```
e.g., ZAT1FO   Col 1        Col 8
                  EBCDIC       360,20,361,21

e.g., ZAT2FO   Col 1        Col 8
                  ASCII        20,360,21,361
```

b. To introduce a user specified 8-bit table:

FIELD (Col)	DESCRIPTION
USER SPECIFIED 8-BIT TABLE (01-74)	Enter the octal value of the input character followed by the octal value of the output character desired. All fields are separated by commas with no intervening blanks. To continue on subsequent cards, terminate the first card with a comma (prior to Col 74) and continue on a subsequent card beginning in Col 1. This option constitutes an entire new transliteration table and an octal character must be provided for each octal value in the data being processed. 256 entries must be specified for ZAT1FO while 64 entries must be specified for ZAT2FO.

```
e.g., Card #1   Col 1
                   0,1,2,3,.............76,77,
      Card #2      100,101,102,...........
```

FIGURE 10-05. Transliteration Card(s)

ZAT2FO LABEL CARD

FIELD (Col)	DESCRIPTION
USER LABEL INFORMATION (01-80)	If user labels are specified, the user must provide label cards. These cards follow all other cards in the input deck. If both header/trailer labels are specified, a header/trailer set should be provided for each reel/file of output expected. If there are more output reels/files than header/trailer sets, the last set will be used on all reels/files exceeding the number of sets provided. If there are more header/trailer sets provided than output reels/files, the last reel/file will contain the correct header; however, the last trailer provided will be used as the final trailer label. If only header labels are specified, header label cards should be provided in the order to be written. If the number of reels/files exceed the number of headers provided, the last header will be written to all subsequent reels/files.

FIGURE 10-06. Label Card

6-BIT TABLES

H200 OCTAL	BCL OCTAL	H6000 OCTAL	H6000 GRAPHIC
00	00	00	0
01	01	01	1
02	02	02	2
03	03	03	3
04	04	04	4
05	05	05	5
06	06	06	6
07	07	07	7
10	10	10	8
11	11	11	9
57	12	12	[
75	13	13	#
15	14	14	@
20	15	15	:
16	16	16	>
32	57	17	?
60	20	20	space
21	21	21	A
22	22	22	B
23	23	23	C
24	24	24	D
25	25	25	E
26	26	26	F
27	27	27	G
30	30	30	H
31	31	31	I
56	32	32	&
33	33	33	
55	12	34	]
74	35	35	(
36	36	36	<
00	77	37	\
52	40	40	^
41	41	41	J
42	42	42	K
43	43	43	L
44	44	44	M
45	45	45	N
46	46	46	O
47	47	47	P
50	50	50	Q
51	51	51	R
13	52	52	-
53	53	53	$
54	54	54	*
76	55	55	)
56	56	56	;
40	57	57	'
36	60	60	+
61	61	61	/
62	62	62	S
63	63	63	T
64	64	64	U
65	65	65	V
66	66	66	W
67	67	67	X

FIGURE 10-07. 6-Bit Tables

H200 OCTAL	BCL OCTAL	H6000 OCTAL	H6000 GRAPHIC
70	70	70	Y
71	71	71	Z
14	72	72	_
73	73	73	,
35	74	74	$
75	75	75	=
76	34	76	"
77	76	77	!

FIGURE 10-07. 6-Bit Tables (cont)

EBCDIC TRANSLITERATION TABLE

EBCDIC		H6000	
Hexadecimal	Octal	Octal	Graphic
00-49	00-111	20	space
4A	112	12	[
4B	113	33	.
4C	114	36	<
4D	115	35	(
4E	116	60	+
4F	117	20	space
50	120	32	&
51-59	121-131	20	space
5A	132	34	]
5B	133	53	$
5C	134	54	*
5D	135	55	)
5E	136	56	;
5F	137	72	¬
60	140	52	-
61	141	61	/
62-6A	142-152	20	space
6B	153	73	:
6C	154	74	%
6D	155	72	_
6E	156	16	>
6F	157	17	(?)
70-79	160-171	20	space
7A	172	15	:
7B	173	13	#
7C	174	14	@
7D	175	57	'
7E	176	75	=
7F	177	76	"
80-BF	200-277	20	space
C0	300	20	space (PZ)
C1	301	21	A
C2	302	22	B
C3	303	23	C
C4	304	24	D
C5	305	25	E
C6	306	26	F
C7	307	27	G
C8	310	30	H
C9	311	31	I
CA-CF	312-317	20	space
D0	320	40	(minus zero)*
D1	321	41	J
D2	322	42	K
D3	323	43	L
D4	324	44	M
D5	325	45	N
D6	326	46	O
D7	327	47	P
D8	330	50	Q
D9	331	51	R

FIGURE 10-08. EBCDIC Transliteration Table

EBCDIC		H6000	
Hexadecimal	Octal	Octal	Graphic
DA-E1	332-341	20	space
E2	342	62	S
E3	343	63	T
E4	344	64	U
E5	345	65	V
E6	346	66	W
E7	347	67	X
E8	350	70	Y
E9	351	71	Z
EA-EF	352-357	20	space
F0	360	00	0
F1	361	01	1
F2	362	02	2
F3	363	03	3
F4	364	04	4
F5	365	05	5
F6	366	06	6
F7	367	07	7
F8	370	10	8
F9	371	11	9
FA-FF	372-377	20	space

FIGURE 10-08. EBCDIC Transliteration Table (cont)

H6000 UASCI TRANSLITERATION TABLE

H6000 UASCI		H6000	
HEXADECIMAL	OCTAL	OCTAL	GRAPHIC
00-20	00-40	20	space
21	41	77	!
22	42	76	"
23	43	13	#
24	44	53	$
25	45	74	%
26	46	32	&
27	47	57	'
28	50	35	(
29	51	55	)
2A	52	54	*
2B	53	60	+
2C	54	73	,
2D	55	52	-
2E	56	33	.
2F	57	61	/
30	60	00	0
31	61	01	1
32	62	02	2
33	63	03	3
34	64	04	4
35	65	05	5
36	66	06	6
37	67	07	7
38	70	10	8
39	71	11	9
3A	72	15	:
3B	73	56	;
3C	74	36	<
3D	75	75	=
3E	76	16	>
3F	77	17	?
40	100	14	@
41	101	21	A
42	102	22	B
43	103	23	C
44	104	24	D
45	105	25	E
46	106	26	F
47	107	27	G
48	110	30	H
49	111	31	I
4A	112	41	J
4B	113	42	K
4C	114	43	L
4D	115	44	M
4E	116	45	N
4F	117	46	O
50	120	47	P
51	121	50	Q
52	122	51	R
53	123	62	S
54	124	63	T
55	125	64	U
56	126	65	V
57	127	66	W

FIGURE 10-09. H6000 UASCI Transliteration Table

H6000 UASCI		H6000	
HEXADECIMAL	OCTAL	OCTAL	GRAPHIC
58	130	67	X
59	131	70	Y
5A	132	71	Z
5B	133	12	[
5C	134	37	\
5D	135	34	]
5E	136	40	^
5F	137	72	-
60-7F	140-177	20	space

FIGURE 10-09. H6000 UASCI Transliteration Table (cont)

H200 EVEN PARITY TAPES

(Also IBM 7030)

Tape Value	Translated Value	Graphic
15	36	<
20	20	space
32	17	(?)
37	56	¦
52	57	#
57	13	+
60	60	-/$ (minus $)
40	40	

NOTE: These are suggested character transliteration for H200/IBM 7080 Even Parity Tapes. However, the user may change any tape input value by entering input and output value in the TYPE 2 parameter card.

FIGURE 10-10. H200 Even Parity Tapes

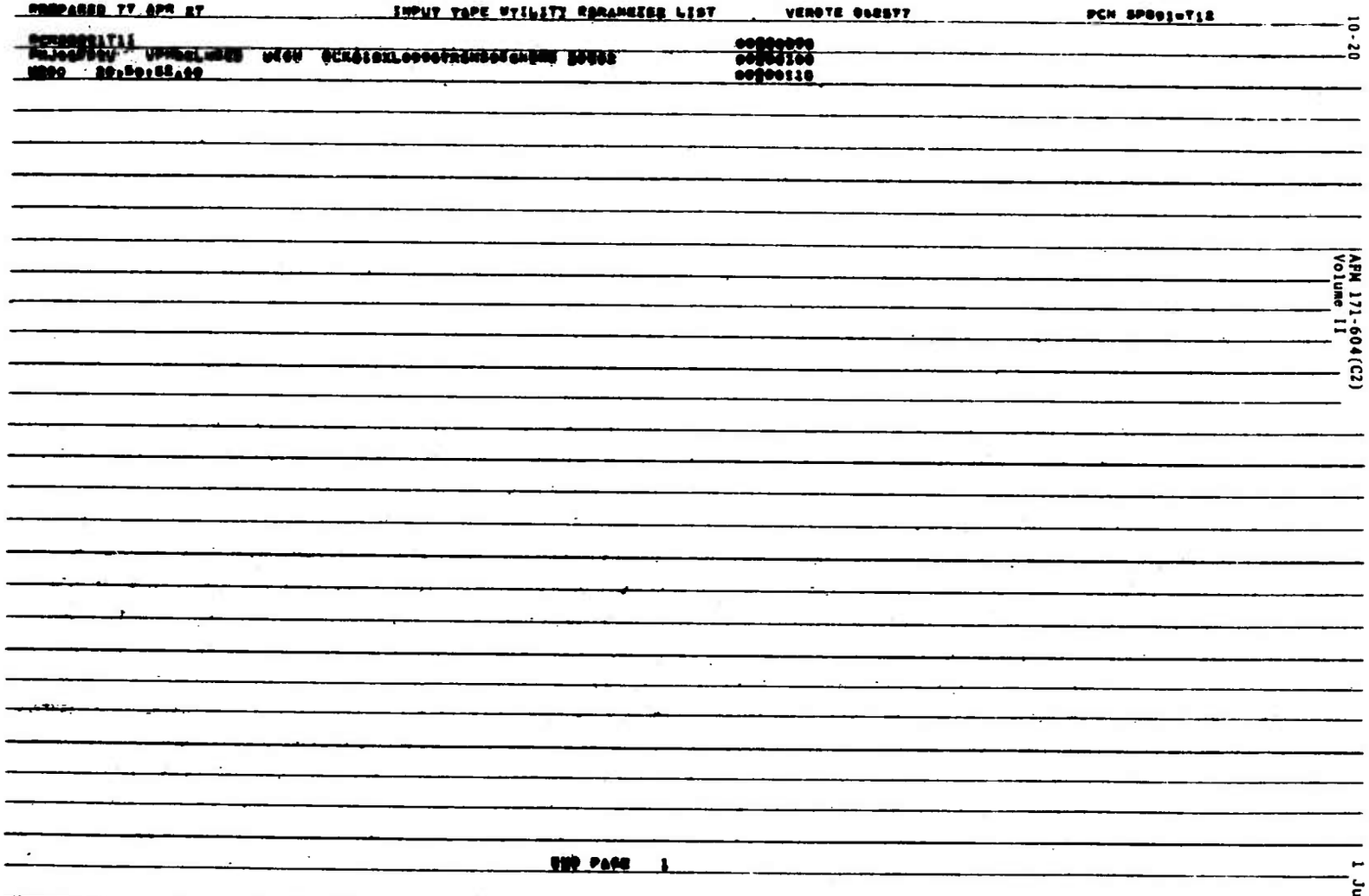
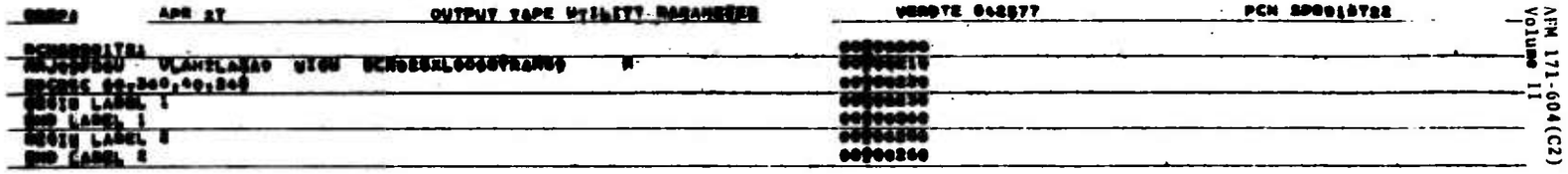


FIGURE 10-11. Input Tape Utility Parameter List



END PAGE 1

FIGURE 10-12. Output Tape Utility Parameter List

PART SIX - B3500 BACKUP PROCESSING

SECTION 11. SYSTEM SUMMARY

11.1 System Application. The purpose of this system is to provide users with the capability to print B3500 Printer Backup or punch B3500 Punch Backup Tapes on the H6000 system.

11.2 System Operation. N/A.

11.3 System Configuration. These programs were written for use on the H6000 Computer System with tape, card reader, card punch and printer.

11.4 System Organization. The system consists of two (2) stand-alone programs. ZAK1FO - H6000 Processing of B3500 Print Backup Tapes and ZAK3FO - H6000 Processing of B3500 Punch Backup Tapes.

11.5 Performance. N/A.

11.6 Data Base. N/A.

11.7 General Descriptions of Inputs, Processing, Outputs:

11.7.1 ZAK1FO:

a. Inputs for ZAK1FO will be parameter card (ref Figure 15-01) and a B3500 Printer Backup Tape that will be considered to have the following characteristics:

-BLOCKED OR UNBLOCKED
-STANDARD LABELS
-9TRACK
-HIGH DENSITY
-FIXED RECORD LENGTH OF 136 EBCDIC CHARACTERS

* b. Output will be a listing that is identical to B3500 listings with the following exceptions:

B3500 Print Character	H6000 Print Character
¢ (cent sign)	(backward slash)
(bar)	(backward slash)
(tilde)	(tilde)
! (exclamation mark)	H6000 is unable to process these characters (results unpredictable)
? (question mark)	

*11.7.2 ZAK3FO:

a. Input for ZAK3FO is a B3500 Punch Backup Tape which must be:

-BLOCKED-1 or BLOCKED-6
-STANDARD LABELS
-9-TRACK
-FIXED RECORD LENGTH OF 136 EBCDIC CHARACTERS

b. Output will be a punched card deck which is identical to the output of the B3500.

SECTION 12 - STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

12.1 ZAK1FO - Processing of B3500 Print Backup Tapes:

12.1.1 Initiation Procedures. To initiate execution of ZAK1FO the user may execute Job Control Language (JCL) ref paragraph 12.1.2.4.

12.1.1.1 Program Notes. Program ZAK1FO reads a B3500 print backup tape, translates the EBCDIC characters into H6000 characters and prints output listings identical to those produced on the B3500. The use of a parameter card and/or $ control card(s) allows the user the following options:

a. Print an entire tape or selectively print any individual file on the tape.

b. If a multi-reel report is to be printed a separate run is required for each reel of tape.

c. Restart option to resume printing if the job is stopped or interrupted. When the Restart option is used, printing starts with the file and page specified and prints the remainder of the file plus any files that follow on the input tape.

d. User can specify part paper desired:

(1) With a $ REPORT control card if output print is assigned to SYSOUT with a $ SYSOUT control card.

(2) By using fields 5 and 6 of the parameter card (ref paragraph 12.1.2.1) and a $ PRINT control card (Not recommended on pre 6.0 H6000 GCOS system release because the report may not be formatted correctly.)

e. Special carriage tape is not required.

12.1.1.2 References:

a. H6000 Reference Manual BS19.

b. H6000 BR38.

c. H6000 GCOS BR43.

d. AFM 171-110, Volume II.

e. Burroughs B2500/3500 COBOL Reference Manual (appendix C).

*12.1.2 Staff Input Requirements. Input is a B3500 Printer Backup Tape that is considered to be:

-UNBLOCKED
-STANDARD LABELS
-9TRACK
-HIGH DENSITY
-FIXED RECORD LENGTH OF 136 EBCDIC CHARACTERS
-CREATED WITH THE "PBTB" OPTION OFF
-MUST BE MCPV GENERATED AND NOT SPOOLED OFF OF DISK BACKUP

12.1.2.1 Input Formats:

FIELD	CARD COL	CONTENTS
PCN	1-18	PCNSS891K11ßßßßßßßß
1	19-25	PRINTßßß or RESTART (see Note 1)
2	27-32	ALLßßß or FILE-ID (see Note 2)
3	34-37	BLANK or PAGE (see Note 3)
4	39-42	BLANK or INTEGER-1 (see Note 4)
5	44-47	BLANK or PART (see Note 5)
6	49	BLANK or INTEGER-2 (see Note 6)
7	50-80	BLANK

Note 1. ß=blank. PRINTßßß will be used in all instances except when restarting a job that has been interrupted. Then RESTART option will be used.

Note 2. Use ALLßßß to print an entire tape. Use the six (6) digit B3500 File-ID to specify the file in which printing is to be restarted when using the RESTART option.

Note 3. Leave blank except when using the RESTART option. For restarting, punch the word PAGE in this field.

Note 4. Leave blank except when using the RESTART option. For restarting, Integer-1 will be a 4-digit right justified page number that you wish to be the first page printed on the restart. If you wish to resume printing with page 89, this field will be ØØ89. Integer-1 must be greater than ØØØ1.

Note 5. Leave blank except to specify that multiple part paper is desired. To use this option, punch the word PART in this field. This option is ignored when printer file is assigned to SYSOUT.

Note 6. Leave blank except when using Field 5 above to specify multiple part paper. The desired part paper is punched in this field (i.e., 2,3,4, etc.)

Note 7. Only one parameter card may be used per activity.

12.1.2.2 Composition Rules. N/A.

12.1.2.3 Input Vocabulary. N/A.

12.1.2.4 Sample Inputs. A sample job stream to print all printer backup files on the tape and using $ REPORT control card to specify part paper desired with output assigned to SYSOUT.

```
Col 1     Col 8       Col 16

    $         IDENT       (User Specified)
    $         USERID      (User Specified)
    $         LIBRARY     LA
    $         USE         ZAK1FO
    $         ENTRY       C.ZAK1
    $         EXECUTE
    $         LIMITS      20,20K
    $         TAPE9       LA,XØD,, nnnnn,,USER-LIBRARY
    $         TAPE9       FA,X1D,,99999,,B3500-PRINTER-BACKUP-TAPE
    $         REPORT      63,PR,3PARTPAPER(see note 1)
    $         SYSOUT      P1
    PCNSS891K11ßßßßßßßßPRINTßßßALLßßß
    $         ENDJOB
```

Note 1. Omit $ REPORT control card if special printer paper is not desired.

12.1.3 Output Requirements. Output will be a B3500 listing.

12.1.3.1 Output Formats. N/A.

12.1.3.2 Sample Outputs. N/A.

12.1.3.3 Output Vocabulary. N/A.

12.1.4 Utilization of System Output. N/A.

12.1.5 Recovery and Error Correction Procedures. N/A.

12.2 ZAK3FO - Processing of B3500 Punch Backup Tapes:

12.2.1 Initiation Procedures. To initiate execution of ZAK3FO the user may execute Job Control Language (JCL) ref. paragraph 12.2.2.4.

* 12.2.1.1 Program Notes. Program ZAK3FO is designed to read punch backup tapes from the B3500 and output punch card decks identical to the B3500 output. The punch is dedicated to the program during execution. Figure 12-01 contains translation tables which provide a cross reference between graphics, B3500 internal code, H6000 octal code and H6000 card codes. Extreme care should be taken to assign the correct File code to coincide with the blocking Factor of the input punch tape (see para 12.2.2.4). Failure to assign propoer File code will result in unpredictable output.

12.2.1.2 References:

a. H6000 Reference Manual BS19.

b. H6000 BR38.

c. H6000 GCOS BR43.

d. AFM 171-110 Volume II.

e. BURROUGHS B2500/3500COBOL Reference Manual (appendix C).

* 12.2.2 Staff Input Requirements. Parameter cards are not required. Reference AFM 171-604 Vol II para 11.7.2a for restrictions on the input tape.

12.2.2.1 Input Formats. N/A.

12.2.2.2 Composition Rules. N/A.

12.2.2.3 Input Vocabulary. N/A.

* 12.2.2.4 Sample Inputs. A sample job stream to punch a card deck from a B3500 Punch Backup Tape:

```
Col 1        Col 8          Col 16

    $            IDENT          (User Specified)
    $            USERID         (User Specified)
    $            LIBRARY        LA
    $            USE            ZAK3F0
    $            ENTRY          ZAK3F0
    $            EXECUTE
    $            LIMITS         ,4K
    $            TAPE9          LA,X##,nnnnn,,USER-LIBRARY
    $            TAPE9          F1,X1D,,nnnnn,,B3500-BLOCKED-1-
                                BACKUP-PUNCH-TAPE (OPTIONAL)
    $            TAPE9          F6,X1D,nnnnn, B3500-Block-6-
                                BACKUP-PUNCH-TAPE (OPTIONAL)
    $            PUNCH          CL
    $            ENDJOB
```

b. One of the two(2) optional files F1 or F6 is required in the above JCL.

12.2.3 Output Requirements. Output will be a punch card deck, identical to the B3500 output.

12.2.3.1 Output Formats. N/A.

12.2.3.2 Sample Outputs. N/A.

12.2.3.3 Output Vocabulary. N/A.

12.2.4 Utilization of System Outputs. N/A.

12.2.5 Recovery and Error Correction Procedures. N/A.

TRANSLATION TABLE

B3500 Graphic	B3500 8-Bit Internal	H6000 Octal	H6000 Card Code
NULL	00	5403	12-0-1-8-9
SOH	01	4401	12-1-9
STX	02	4201	12-2-9
ETX	03	4101	12-3-9
	04	4041	12-4-9
HT	05	4021	12-5-9
	06	4011	12-6-9
DEL	07	4005	12-7-9
	08	4003	12-8-9
	09	4403	12-1-8-9
	0A	4203	12-2-8-9
VT	0B	4103	12-3-8-9
FF	0C	4043	12-4-8-9
CR	0D	4023	12-5-8-9
SO	0E	4013	12-6-8-9
SI	0F	4007	12-7-8-9
DLE	10	6403	12-11-1-8-9
DC1	11	2401	11-1-9
DC2	12	2201	11-2-9
DC3	13	2101	11-3-9
	14	2041	11-4-9
NL	15	2021	11-5-9
BS	16	2011	11-6-9
	17	2005	11-7-9
CAN	18	2003	11-8-9
EM	19	2403	11-1-8-9
	1A	2203	11-2-8-9
	1B	2103	11-3-8-9
FS	1C	2043	11-4-8-9
GS	1D	2023	11-5-8-9
RS	1E	2013	11-6-8-9
US	1F	2007	11-7-8-9
	2Ø	3403	11-0-1-8-9
	21	1401	0-1-9
	22	1201	0-2-9
	23	1101	0-3-9
	24	1041	0-4-9
LF	25	1021	0-5-9
ETB	26	1011	0-6-9
ESC	27	1005	0-7-9
	28	1003	0-8-9
	29	1403	0-1-8-9
	2A	1203	0-2-8-9
	2B	1103	0-3-8-9
	2C	1043	0-4-8-9
ENQ	2D	1023	0-5-8-9
ACK	2E	1013	0-6-8-9
BEL	2F	1007	0-7-8-9
	30	7403	12-11-0-1-8-9
	31	0401	1-9
SYN	32	0201	2-9
	33	0101	3-9
	34	0041	4-9

FIGURE 12-01. Translation Table

B3500 Graphic	B3500 8-Bit Internal	H6000 Octal	H6000 Card Code
	35	0021	5-9
	36	0011	6-9
EOT	37	0005	7-9
	38	0003	8-9
	39	0403	1-8-9
	3A	0203	2-8-9
	3B	0103	3-8-9
DC4	3C	0043	4-8-9
NAK	3D	0023	5-8-9
	3E	0013	6-8-9
SUB	3F	0007	7-8-9
SPACE	40	0000	
	41	5401	12-0-1-9
	42	5201	12-0-2-9
	43	5101	12-0-3-9
	44	5041	12-0-4-9
	45	5021	12-0-5-9
	46	5011	12-0-6-9
	47	5005	12-0-7-9
	48	5003	12-0-8-9
	49	4402	12-1-8
[	4A	4202	12-2-8
.	4B	4102	12-3-8
	4C	4042	12-4-8
(	4D	4022	12-5-8
+	4E	4012	12-6-8
!	4F	4006	12-7-8
&	50	4000	12
	51	6401	12-11-1-9
	52	6201	12-11-2-9
	53	6101	12-11-3-9
	54	6041	12-11-4-9
	55	6021	12-11-5-9
	56	6011	12-11-6-9
	57	6005	12-11-7-9
	58	6003	12-11-8-9
	59	2402	11-1-8
	5A	2202	11-2-8
$	5B	2102	11-3-8
*	5C	2042	11-4-8
)	5D	2202	11-5-8
;	5E	2012	11-6-8
	5F	2006	11-7-8
-	60	2000	11
/	61	1400	0-1
	62	3201	11-0-2-9
	63	3101	11-0-3-9
	64	3041	11-0-4-9
	65	3021	11-0-5-9
	66	3011	11-0-6-9
	67	3005	11-0-7-9
	68	3003	11-0-8-9
	69	1402	0-1-8
	6A	6000	12-11
	6B	1102	0-3-8
¢	6C	1042	0-4-8
	6D	1022	0-5-8
¬	6E	1012	0-6-8

FIGURE 12-01. Translation Table (cont)

B3500 Graphic	B3500 8-Bit Internal	H6000 Octal	H6000 Card Code
?	6F	1006	0-7-8
	70	7000	12-11-0
	71	7401	12-11-0-1-9
	72	7201	12-11-0-2-9
	73	7101	12-11-0-3-9
	74	7041	12-11-0-4-9
	75	7021	12-11-0-5-9
	76	7011	12-11-0-6-9
	77	7005	12-11-0-7-9
	78	7003	12-11-0-8-9
	79	0402	1-8
:	7A	0202	2-8
#	7B	0102	3-8
@	7C	0042	4-8
'	7D	0022	5-8
=	7E	0012	6-8
"	7F	0006	7-8
	80	5402	12-0-1-8
a	81	5400	12-0-1
b	82	5200	12-0-2
c	83	5100	12-0-3
d	84	5040	12-0-4
e	85	5020	12-0-5
f	86	5010	12-0-6
g	87	5004	12-0-7
h	88	5002	12-0-8
i	89	5001	12-0-9
	8A	5202	12-0-2-8
	8B	5102	12-0-3-8
	8C	5042	12-0-4-8
	8D	5022	12-0-5-8
	8E	5012	12-0-6-8
	8F	5005	12-0-7-8
	90	6402	12-11-1-8
j	91	6400	12-11-1
k	92	6200	12-11-2
l	93	6100	12-11-3
m	94	6040	12-11-4
n	95	6020	12-11-5
o	96	6010	12-11-6
p	97	6004	12-11-7
q	98	6002	12-11-8
r	99	6001	12-11-9
	9A	6202	12-11-8-2
	9B	6102	12-11-8-3
	9C	6042	12-11-8-4
	9D	6022	12-11-8-5
	9E	6012	12-11-8-6
	9F	6006	12-11-8-7
	AØ	3402	11-0-8-1
	A1	3400	11-0-1
s	A2	3200	11-0-2
t	A3	3100	11-0-3
u	A4	3040	11-0-4
v	A5	3020	11-0-5
w	A6	3010	11-0-6
x	A7	3004	11-0-7

FIGURE 12-01. Translation Table (cont)

B3500 Graphic	B3500 8-Bit Internal	H6000 Octal	H6000 Card Code
	A8	3002	11-0-8
y	A9	3001	11-0-9
z	AA	3202	11-0-2-8
	AB	3102	11-0-3-8
	AC	3042	11-0-4-8
	AD	3022	11-0-5-8
	AE	3012	11-0-6-8
	AF	3006	11-0-7-8
	BØ	7402	12-11-0-1-8
	B1	7400	12-11-0-1
	B2	7200	12-11-0-2
	B3	7100	12-11-0-3
	B4	7040	12-11-0-4
	B5	7020	12-11-0-5
	B6	7010	12-11-0-6
	B7	7004	12-11-0-7
	B8	7002	12-11-0-8
	B9	7001	12-11-0-9
	BA	7202	12-11-0-2-8
	BB	7102	12-11-0-3-8
	BC	7042	12-11-0-4-8
	BD	7022	12-11-0-5-8
	BE	7012	12-11-0-6-8
	BF	7006	12-11-0-7-8
(+)PZ	CØ	5000	12-0
A	C1	4400	12-1
B	C2	4200	12-2
C	C3	4100	12-3
D	C4	4040	12-4
E	C5	4020	12-5
F	C6	4010	12-6
G	C7	4004	12-7
H	C8	4002	12-8
I	C9	4001	12-9
	CA	5203	12-0-2-8-9
	CB	5103	12-0-3-8-9
	CC	5043	12-0-4-8-9
	CD	5023	12-0-5-8-9
	CE	5013	12-0-6-8-9
	CF	5007	12-0-7-8-9
(!)MZ	DØ	3000	11-0
J	D1	2400	11-1
K	D2	2200	11-2
L	D3	2100	11-3
M	D4	2040	11-4
N	D5	2020	11-5
O	D6	2010	11-6
P	D7	2004	11-7
Q	D8	2002	11-8
R	D9	2001	11-9
	DA	6203	12-11-2-8-9
	DB	6103	12-11-3-8-9
	DC	6043	12-11-4-8-9
	DD	6023	12-11-5-8-9
	DE	6013	12-11-6-8-9
	DF	6007	12-11-7-8-9
	EØ	1202	0-2-8
	E1	3401	11-0-1-9

FIGURE 12-01 Translation Table (cont)

B3500 Graphic	B3500 8-Bit Internal	H6000 Octal	H6000 Card Code
S	E2	1200	0-2
T	E3	1100	0-3
U	E4	1040	0-4
V	E5	1020	0-5
W	E6	1010	0-6
X	E7	1004	0-7
Y	E8	1002	0-8
Z	E9	1001	0-9
	EA	3203	11-0-2-8-9
	EB	3103	11-0-3-8-9
	EC	3043	11-0-4-8-9
	ED	3023	11-0-5-8-9
	EE	3013	11-0-6-8-9
	EF	3007	11-0-7-8-9
ø	Fø	1000	ø
1	F1	0400	1
2	F2	0200	2
3	F3	0100	3
4	F4	0040	4
5	F5	0020	5
6	F6	0010	6
7	F7	0004	7
8	F8	0002	8
9	F9	0001	9
	FA	7203	12-11-0-2-8-9
	FB	7103	12-11-0-3-8-9
	FC	7043	12-11-0-4-8-9
	FD	7023	12-11-0-5-8-9
	FE	7013	12-11-0-6-8-9
	FF	7007	12-11-0-7-8-9

FIGURE 12-01. Translation Table (cont)

PART SEVEN - TEST FILE GENERATOR PROGRAMS

SECTION 13. SYSTEM SUMMARY

13.1 System Application. The purpose of the test file generator utility is to enable programmers to create test data files for subsequent use as input to a designated program. Program development often requires that test data files be created for adequate validation. This utility enables data bases to be created for this purpose. Input is accepted from cards, disk or tape and placed on disk or tape in standard system format. The size of the input and output is designated via a control card.

13.2 System Operation. N/A.

13.3 System Configuration. The test file generator program was written for use on the H6000 computer system with tape and disk I/O devices.

13.4 System Organization. The test file generator utility consists of two (2) stand-alone programs (ZACØFO and ZADØFO).

13.4.1 ZACØFO - Test File Generator From Cards. ZACØFO uses punched cards as input to produce a SSF output file.

13.4.2 ZADØFO - Test File Generator From Tape/Disk. ZADØFO uses tape or disk as input to produce a SSF output file.

13.5 Performance. N/A.

13.6 Data Base. The test file generator programs support card (ZACØFO) and tape or disk (ZADØFO) data files that are in SSF.

13.7 General Description of Inputs, Processing, and Outputs:

13.7.1 Inputs. A control card followed by either punched cards or any standard system formatted disk or tape file in which the user desires to change the number of characters per record.

13.7.2 Processing. The test file generator programs provide the capability to process input that consists of a specified number of characters per record and create an output file containing a different number of characters per record. The change in record size is supplied to the program by a control card.

13.7.3 Output. A disk or tape file in SSF with an adjusted number of characters per record.

SECTION 14. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

14.1 ZAC$FO - Test File Generator From Cards. The function of this program is to create a data file on disk or tape from punched cards.

14.1.1 Initiation Procedures:

14.1.1.1 Program Notes:

a. Only one file may be created each time the program is executed.

b. Library tape LA is in sequential R* format.

14.1.1.2 References. H6000 Control Card Manual, BS19.

14.1.2 Staff Input Requirements. Input consists of a control card followed by card data. User requirements will determine frequency of processing.

14.1.2.1 Input Formats:

a. PCN Card:

Column	Contents	Purpose
1-11	PCNSP891C$1	Deck Identifier
12-8$	Blank	

b. Control Card:

Column	Contents	Purpose
1	*	Card Identifier
2-6	XXXXX	Number of input characters per logical record.
7-12	XXXXXX	Number of output characters per logical record.
13-14	XX	File code of output file (F1).
15-8$	Blank	

c. Data. No specific format required on the punched cards.

14.1.2.2 Composition Rules:

a. Control Card:

(1) An asterisk is required in cc one of the control card.

(2) The size of the input record must range between 1 and 19$8 characters.

(3) If the input character count is omitted from the control card, the default will be 8$ characters.

(4) The size of the output records must range between 1 and 19$8 characters.

(5) If the output character count is omitted from the control card, the default will be 84 characters.

(6) If the input size is less than the output size, the difference will be padded with zeros. For example, if only 4Ø characters of the 8Ø on a data card were desired for the output file and the output size was 84, character positions 41 through 84 would contain zeros. The control card format for this example would be *ØØØ4ØØØØØ84F1.

(7) If output size is less than input size and not a multiple of a 6 (word size), the remainder of the last partially filled word will be padded with zeros. For instance, if input size is 8Ø and output size is 56 characters, a record of 6Ø characters would be written to the output file with the last four characters containing zeros. Positions 1-56 would be used from the input record.

(8) The file code on the control card must match the file code on the appropriate JCL card (F1).

14.1.2.3 Input Vocabulary. N/A.

14.1.2.4 Sample Inputs. To create SSF disk or tape data files, the user may execute the following JCL:

```
Col 1        Col 8          Col 16
    $            IDENT          (USER SPECIFIED)
    $            USERID         (USER SPECIFIED)
    $            LIBRARY        LA
    $            USE            ZACØFO
    $            ENTRY          ZACØFO
    $            EXECUTE
    $            LIMITS         ,4K
    $            TAPE9          LA,X1D,,nnnnn,,USER-LIB
    $            PRMFL          F1,W,S,USERID/FILE NAME
    $            DATA           I*
    PCNSP891CØ1
    *ØØØ8ØØØØØ84F1
    (CARD DATA)
    $            ENDJOB
```

14.1.3 Output Requirements. Output will be a SSF file containing the number of characters per record that were specified on the control card.

14.1.3.1 Output Formats. System Standard Format.

14.1.3.2 Sample Outputs. N/A.

14.1.3.3 Output Vocabulary. N/A.

14.1.4 Utilization of System Outputs. User determined.

14.1.5 Recovery and Error Correction Procedures. N/A.

14.2 ZADØFO - Test File Generator From Tape/Disk. The function of this program is to create a data file on disk or tape from disk or tape input.

14.2.1 Initiation Procedures. To create disk or tape data.

14.2.1.1 Program Notes:

a. Only one file may be created each time the program is executed.

b. Library tape LA is in sequential R* format.

*

14.2.1.2 References:

a. H6000 Control Card Manual, BS19.

b. H6000 File and Record Control Manual, DDØ7.

14.2.2 Staff Input Requirements. Input consists of a control card and data on disk or tape. User requirements will determine frequency of processing.

14.2.2.1 Input Formats:

a. PCN Card:

Column	Contents	Purpose
1-11	PCNSP89IDØ1	Deck Identifier
12-8Ø	Blank	

b. Control Card.

Column	Contents	Purpose
1	*	Card Identifier
2-6	XXXXX	Number of input characters per logical record.
7-12	XXXXXX	Number of output characters per logical record.
13-14	XX	File code of output file (F1).
15-80	Blank	

c. Data. Standard system format is required.

14.2.2.2 Composition Rules:

a. Control Card:

(1) An asterisk is required in column one of the control card.

(2) The size of the input record must range between 1 and 19Ø8 characters.

(3) If the input character count is omitted from the control card, the default will be 8Ø characters.

(4) The size of the output records must range between 1 and 19Ø8 characters.

(5) If the output character count is omitted from the control card, the default will be 84 characters.

(6) If the input size is less than the output size, the difference will be padded with zeros. For example, if only 4Ø characters of the 8Ø on a data card were desired for the output file and the output size was 84, character positions 41 through 84 would contain zeros. The control card format for this example would be *ØØØ4ØØØØØ84F1.

(7) If output size is less than input size and not a multiple of 6 (word size), the remainder of the last partially filled word will be padded with zeros. For instance, if input size is 8Ø and output size is 56 characters, a record of 6Ø characters would be written to the output file with the last four characters containing zeros. Positions 1-56 would be used from the input record.

(8) The file code on the control card must match the file code on the appropriate JCL card (F1).

14.2.2.3 Input Vocabulary. N/A.

14.2.2.4 Sample Inputs. To create a SSF file with 40 character record input and 84 character output:

```
Col 1          Col 8            Col 16
    $              IDENT            (USER SPECIFIED)
    $              USERID           (USER SPECIFIED)
    $              LIBRARY          LA
    $              USE              ZAD0FO
    $              ENTRY            ZAD0FO
    $              EXECUTE
    $              LIMITS           ,4K
    $              TAPE9            LA,X0D,,nnnnn,,USER-LIB
    $              TAPE9            F2,X2D,,nnnnn,,TEST-DATA-INPUT
    $              TAPE9            F1,X1D,,99999,,ZAD0FO-OUTPUT
    $              DATA             I*
    PCNSP891D01
    *0004000084F1
    $              ENDJOB
```

14.2.3 Output Requirements. Output will be a SSF file containing the number of characters per record that were specified on the control card.

14.2.3.1 Output Formats. N/A.

14.2.3.2 Sample Outputs. N/A.

14.2.3.3 Output Vocabulary. N/A.

14.2.4 Utilization of System Outputs. User determined.

14.2.5 Recovery and Error Correction Procedures. N/A.

PART EIGHT - COMMAND IDENTIFIER SUBROUTINE

SECTION 15. SYSTEM SUMMARY

15.1 System Application. The purpose of this subroutine is to eliminate the necessity of changing many standard programs if the location of the command code in the $IDENT control card is changed. It moves the command code from the slave prefix area to a user work area for output label processing, file control table file-ID modification, or other uses required.

15.2 System Operation. N/A.

15.3 System Configuration. The Command Identifier Subroutine was written for H6000 computer systems with tape and disk I/O devices.

15.4 System Organization. The Command Identifier is a GMAP coded subroutine (ZAP7FO) that is called by a user program.

15.4.1 ZAP7FO - Command Identifier Subroutine. ZAP7FO moves the command code from the $IDENT control card image within the slave prefix area to an area specified by the calling program.

15.5 Performance. N/A.

15.6 Data Base. N/A.

15.7 General Description of Inputs, Processing, Outputs:

15.7.1 Input. N/A.

15.7.2 Processing. The Command Identifier Subroutine provides sites, who support other commands, the ability to differentiate between command data without changing standard programs when the command code in the $IDENT is relocated.

15.7.3 Outputs. N/A.

SECTION 16. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

16.1 ZAP7FO - Command Identifier Subroutine. The function of this subroutine is to move the command code from the slave prefix area to a user work area for output label processing, file control table file-ID modification, or other uses.

16.1.1 Initiation Procedures. The subroutine ZAP7FO must be available to the general loader on an object library to be loaded with the calling program. The object library may be made available to the loader by using the file code *L on a file control card which allocates the library or by using a $LIBRARY control card.

16.1.1.1 Program Notes:

a. ZAP7FO is referenced with a call statement or a call pseudo operation:

In COBOL - call ZAP7FO using L1
In GMAP - call ZAP7FO (L1)

The argument 'L1' is a symbolic name of a location in core.

b. The subroutine will move one of two fields from the $IDENT card to the first two character positions of the first word of the location specified in the argument of the call statement (L1). The field to be moved is determined as follows:

(1) Cc 22-23, if cc 16 of the ident card contains an "F".

(2) Cc 71-72, if cc 16 of the ident card does not contain an "F".

c. The following are coding examples for programs written in COBOL and GMAP.

(1) Example 1 - COBOL.

```
$COBOL
 IDENTIFICATION DIVISION.
    .
    .
 WORKING-STORAGE SECTION.
 77    L1     PIC XX VALUE IS SPACES.
    .
    .
 PROCEDURE DIVISION.
    .
    .
 CALL ZAP7FO USING L1.
 MOVE L1 TO CMD-ID-HOLD.
    .
    .
 STOP RUN.
```

(2) Example 2 - GMAP.

```
$GMAP
   .
   .
L1 BCI       1,
   .
   .
 CALL ZAP7FO (L1)
 LDA         L1
 STA         CMD-ID
   .
   .
 END
```

16.1.1.2 References:

a. H6000 General Loader Manual, DD10.

b. H6000 COBOL Reference Manual, BS08.

c. H6000 GMAP Manual, DD08.

d. H6000 Control Card Reference Manual, DD31.

16.1.2 Staff Input Requirements. N/A.

16.1.2.1 Input Formats. N/A.

16.1.2.2 Composition Rules. N/A.

16.1.2.3 Input Vocabulary. N/A.

16.1.2.4 Sample Inputs:

a. The following job control statements are required when using the object library:

```
Col 1       Col 8          Col 16
$           IDENT
$           USERID
              .
              .
$           EXECUTE
              .
              .
$           PRMFL          *L,R,R,CAT/FILE with 7AP7FO
```

b. The following job control statements are required when using a $LIBRARY control card:

```
Col 1       Col 8          Col 16
$           IDENT
$           USERID
              .
              .
$           LIBRARY        LB - See control cards reference manual
              .                   for format and placement
              .
              .
              .
$           EXECUTE
$           PRMFL          LB,R,R,CAT/FILE with ZAP7FO
```

16.1.3 Output Requirements. N/A.

16.1.3.1 Output Formats. N/A.

16.1.3.2 Sample Outputs. N/A.

16.1.3.3 Output Vocabulary. N/A.

16.1.4 Utilization of System Outputs. N/A.

16.1.5 Recovery and Error Correction Procedures. N/A.

PART NINE - COBOL SORT ROUTINE

SECTION 17. SYSTEM SUMMARY

17.1 System Application. The purpose of this COBOL Sort Routine is to reduce core usage in COBOL programs using the SORT verb. The operational improvements provided by this COBOL Sort routine includes reduced core requirement at run time which improves system throughput. This subroutine works in conjunction with the standard COBOL Sort subroutines to allow the Sort/Merge program to use work space which would not otherwise be used. This routine cannot be used in the Merge portion of the H6000 Sort/Merge routines.

17.2 System Operation. N/A.

17.3 System Configuration. The COBOL Sort Routine was written for use on the H6000 computer system with disk I/O devices.

17.4 System Organization. The COBOL Sort Routine consists of one (1) subroutine (ZAPØFO).

17.5 Performance. N/A.

17.6 Data Base. N/A.

17.7 General Descriptions of Inputs, Processing, Outputs:

17.7.1 Inputs. N/A.

17.7.2 Processing. The COBOL Sort Routine is able to reduce the core needed for a COBOL program containing a Sort by using the area containing the File Control Blocks, which is not normally used during a sort, for the sort work area. The contents of the File Control Blocks are moved out to a temporary (H#) file while the sort takes place. After the sort, or if an abort takes place, the File Control Blocks are restored from the H# file. The COBOL Sort Routine is called into the COBOL program by a "CALL PRESTO" statement.

17.7.3 Outputs. N/A.

SECTION 18. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

18.1 ZAPØFO - COBOL Sort Routine (PRESTO). The function of this subroutine is to reduce core requirements at run time for COBOL programs using the SORT verb.

18.1.1 Initiation Procedures. The following example illustrates how ZAPØFO is used:

```
Col 1          Col 8            Col 16
    $              OPTION           COBOL
    $              USE              ZAPØFO
    $              USE              .SMA/1/,.XBUF1/454Ø/
    $              USE              .XBUF2/2/,F1/872/,F3/872/
    $              USE              P1/88Ø/,.SMC/1/
    $              COBOL            NDECK
    (COBOL SOURCE PROGRAM)
                    .
                    .
    Ø5ØØØØ         PROCEDURE DIVISION.
    Ø5Ø1ØØ         ØØ1-BEGIN.
    Ø5Ø2ØØ         CALL PRESTO
                    .
                    .
    Ø55ØØØ         SORT SORT-FILE...
                    .
                    .
    Ø8ØØØØ         STOP RUN.
    $              EXECUTE
    $              LIMITS           ...
    $              PRMFL            *L,R,S,...
    $              FILE             H#,,3R
    $              FILE             S1,,2ØR
    $              DATA             F1
                    .
                    .
                    .
    $              TAPE             F2...
    $              TAPE             F3...
    $              SYSOUT           P1
```

18.1.1.1 Program Notes:

a. This subroutine can not be used with the Merge function of the Sort/Merge routines.

b. ZAPØFO can be used with segmented or non-segmented COBOL programs and is accessed by inserting a "CALL PRESTO" in the COBOL program before the SORT verb.

c. The calling program must have at least one overlayable labeled common area. A labeled common area will be non-overlayable for one of two reasons. First, it may be a file control table (FCT) for a USING or GIVING file, or a SORT file. This particular area must remain non-overlayable. Second, statements may be a part of an INPUT or OUTPUT PROCEDURE which cause the contents of the area to be referenced or changed. In this case, the area can be made overlayable in one of two ways.

(1) If the statements causing a labeled common area to be non-overlayable are not logically an integral part of the processing of sort input or output records, then they can be moved outside the sort input and/or output procedures. Doing this will make the area overlayable.

(2) If any of the statements are logically an integral part of the processing of sort input or output records, then the only way to make the area overlayable is to eliminate the corresponding input and/or output procedures. An input procedure can be eliminated by replacing its RELEASE statements with WRITE statements to a work file; replacing the INPUT PROCEDURE phrase with a USING phrase specifying the work file; and PERFORMing the procedure just before doing the sort. An output procedure can be eliminated in similar fashion. This technique involves a trade-off. It results in extra I/O time approximately equivalent to two passes on the sort file for each input or output procedure eliminated. Parallel test runs will show the implications of this change in each case.

d. ISP Buffers. ISP provides the user with the option to explicitly allocate core for its required buffer pool using labeled common (labels .XBUF1 and .XBUF2). This buffer pool will be overlayable if no CALLs to ISP subroutines are a part of any input or output procedure. The size of .XBUF1 is selected by the user to provide adequate core for the particular ISP application. Refer to para 18.1.1 for a typical job setup.

e. COBOL File Control Tables (FCTs). For each file defined in the FILE Section, COBOL establishes a labeled common area (label equals file code) which contains the file control block record processing area, all buffers and other control information for the file. This combination is called the FCT for the file. The FCT of a file is overlayable if the file is not a sort file; is not referenced in a USING or GIVING phrase; and is not referenced in an OPEN, CLOSE, READ, or WRITE statement which is part of an input or output procedure.

f. Defining the Overlay Area. The sort requires its work space to be a single block of core of sufficient size, usually 6-10K. After the labeled common areas which are to be used for sort space (i.e., FCTs, ISP buffers) have been identified, it is necessary to insure that they form a contiguous block when the program is executing. If the labeled common areas do not comprise a block of adequate size, a filler area must be appended. Finally, the sort must be informed that these areas constitute its work space. In defining the overlay area, all of these things are accomplished. The $ USE control card provides the means of defining this overlay area.

g. The $ USE Card. The following remarks on the General Loader are a necessary introduction to the $ USE card.

(1) The General Loader allows any program to reference symbols (i.e., SYMREF) which are defined (SYMDEFed) within other programs. These references are stored in a table (load table) which is used during the library search for these other programs, now called subprograms. If found, they are loaded into core.

(2) The Loader also allows any program (or subprogram) to reference one or more named core areas (labeled common). Such a reference includes a label and size (in words). The Loader uses the first such reference it encounters to define the labeled common area; a block of core of the specified size is allocated, and its beginning address is associated with the specified name. All such references allow the referencing programs (or subprograms) to load data into the block, and to reference and alter its contents during execution.

(3) The $ USE card is a Loader control card which generates references to external symbols and labeled common. When the Loader encounters a $ USE card, it adds the referenced external symbols to its load table. The referenced labeled common which had not been previously referenced is immediately defined.

h. Defining the Overlay Area With the $ USE Card. Refer to para 18.1.1 for a sample job stream for a COBOL program using ZAPØFO. In this example, the cards define the overlay area.

```
Col 1       Col 8          Col 16
  $           USE            .SMA/1/,.XBUF1/454Ø/
  $           USE            .XBUF2/2/,F1/872/,F3/872/
  $           USE            P1/88Ø/,.SMC/1/
```

These cards must appear before the $ COBOL card so that they will be the first to reference (and therefore, define) the specified labeled common areas. The Loader will then allocate these areas in a contiguous block, .SMA first, .XBUF1 second, etc., and .SMC last.

(1) The sort is designed so that if the special labeled common areas .SMA and .SMC are defined by the user, it will use the block of core, beginning with the first work in .SMA and ending with the first word in .SMC, for work space. Defining .SMA first and .SMC last insures that all labeled common areas defined in between will be used by the sort.

(2) The decimal numbers appearing between slashes represent the size (in words) of the labeled common areas. The total size of the block of core used by the sort is the sum of these sizes. In this example, ISP buffers (.XBUF1, .XBUF2) and the FCTs of three files (F1, F3, P1) are included in the overlay area. The size of the FCTs can be determined in the following manner. Following each COBOL program listing are two or three GMAP summary pages. Page 1 of this summary contains information on labeled common areas referenced by the COBOL program under the subheading "BLOCK". The user will find following this subheading a list of all these areas by label, and adjacent to each label, under the subheading "LENGTH" the size of the area in octal. The labels of FCTs are the same as their corresponding file codes. The sizes found adjacent to these labels are converted to decimal and the decimal value is used in the $ USE card.

(3) The minimum size of the ISP buffer space must be determined by the user based on the extent of the program's ISP activity. (Reference ISP Manual, DD38). In the example, 4542 words (454Ø+2) were allocated for ISP buffers. The minimum ISP buffer space required may have been much less than this, but 4542 words were used so that the sort would have a total of 7K for work space (1+454Ø+2+872+872+88Ø+1 = 7168 = 7K).

(4) If in this example, there was no ISP activity, then the areas .XBUF1 and .XBUF2 would not be present in the overlay area definition. If the sort was still to be allocated 7K, then 4542 words of fill space would have to be appended to the overlay area. The user can do this by adding 4542 to the size of .SMA. The overlay area definition would then look like this:

```
Col 1       Col 8          Col 16
  $           USE            .SMA/4543/,F1/872/
  $           USE            F3/872/,P1/88Ø/,.SMC/1/
```

i. Changes to the COBOL Program. The only required modification to the COBOL program is the addition of the statement "CALL PRESTO". This CALL must be executed before the first sort. It need be executed only once, regardless of the number of sorts in the program, but multiple executions are permissible. In addition, the user may make modifications as discussed in the previous section to cause labeled common areas to be overlayable.

j. Job Control Lanuguage (JCL). Additional JCL cards are required for four purposes:

(1) To cause ZAPØFO to be loaded with the COBOL program, the following control card must be included in the JCL deck:

```
$          USE          ZAPØFO
```

This card should follow the $ OPTION COBOL card and precede any $ COBOL, $ OBJECT, or $ EXECUTE cards.

(2) A temporary random overlay file must be allocated using the following control card:

```
$          FILE          H#,,3R
```

The file size given is typical; however, the size may be increased or decreased as appropriate. The card must follow the $ EXECUTE card.

(3) The library on which ZAPØFO resides must be allocated to the execution activity. The procedure for accomplishing this is determined at the respective installation. A typical way would be to include a card such as the following after the $ EXECUTE card:

```
$          PRMFL          *L,R,S,catalog-file-string
```

Systems Software personnel should be consulted to determine the appropriate library control card to access this subroutine.

(4) The $ USE card is used to define the overlay area. One or more cards may be used. The card(s) should be located in the deck similar to the $ USE ZAPØFO card.

k. Abnormal Termination During a Sort. In the event of program abort, both the COBOL program and ISP execute wrap-up routines which flush buffers, close files, etc. If ZAPØFO is used with the program and the abort occurred during a sort, these areas will not necessarily all be in core. To avoid incomplete and aborted wrap-ups, ZAPØFO intercepts the wrap-up transfer so that the overlay area can be restored prior to the usual wrap-up processing. This is done only if the abort occurred during a sort.

18.1.1.2 References:

a. H6000 Control Card Reference Manual, BS19.

b. H6000 General Loader Manual, DD10.

c. H6000 ISP Manual, DD38.

18.1.2 Staff Input Requirements. N/A.

18.1.2.1 Input Formats. N/A.

18.1.2.2 Composition Rules. N/A.

18.1.2.3 Input Vocabulary. N/A.

18.1.2.4 Sample Inputs. N/A.

18.1.3 Output Requirements. N/A.

18.1.3.1 Output Formats. N/A.

18.1.3.2 Sample Outputs. N/A.

18.1.3.3 Output Vocabulary. N/A.

18.1.4 Utilization of System Outputs. N/A.

18.1.5 Recovery and Error Correction Procedures. N/A.

PART TEN - COBOL AIDS

SECTION 19. SYSTEM SUMMARY

19.1 System Application. The purpose of the B3500 to H6000 COBOL translator aids is to give the user the ability to translate B3500 COBOL programs to H6000 COBOL for use on the H6000 computer systems.

19.2 System Operation. N/A.

19.3 System Configuration. The B3500 to H6000 COBOL translator is written for use on the H6000 computer system with card, tape and disk I/O devices.

19.4 System Organization. The system consists of one (1) stand-alone program (ZABUFO).

19.4.1 ZABUFO - B3500 to H6000 COBOL Translator. Translates programs written in B3500 COBOL to H6000 COBOL.

19.5 Performance. N/A.

19.6 Data Base. N/A.

19.7 General Description of Inputs, Processing, Outputs. See specific information of inputs, processing, and outputs for ZABUFO in Section 20.

SECTION 20. STAFF FUNCTIONS RELATED to TECHNICAL OPERATIONS

20.1 ZABUFO - B3500 to H6000 COBOL Translator. The function of this program is to translate programs written in B3500 COBOL to H6000 COBOL.

20.1.1 Initiation Procedures. To initiate execution of ZABUFO, the user may execute the Job Control Language (JCL) in paragraph 20.1.2.4.

20.1.1.1 Program Notes:

a. The source output is placed on tape or disk under JCL control. If disk is used, appropriate user action must be taken to allocate Catalog/File space prior to the translation.

b. If the source deck used as input contains (11) (Ø) and/or (12) (Ø) punches, program ZAAØFO should be executed as the first activity in the job stream. (Ref AFM 171-604, Part Four, Volumes I and II.)

c. Tape input to the translator must be converted to H6000 system standard format via program ZAT1FO. (Ref AFM 171-604, Part Five, Volumes I and II.)

d. Source program changes made by the translator are:

(1) General Changes:

The change numbers reflected in cc 79-80 of the input source are retained on the appropriate translated source cards.

The translated source cards are renumbered starting with ØØØ1ØØ and incremented by ØØØ1ØØ.

Non-numeric literals are scanned for quotes (") imbedded in the literal. If found, the quote(s) are transliterated to colon(s) and flagged (D16)

(2) IDENTIFICATION DIVISION:

In PROGRAM-ID the program is replaced with the new ID field of the parameter card. (Ref Figure 20.01.)

The remainder of this Division is copied.

(3) ENVIRONMENT DIVISION:

(a) CONFIGURATION SECTION.

1 SOURCE-COMPUTER.

The old paragraph is replaced with the entry 6ØØØ.

2 OBJECT-COMPUTER.

B3500 unique information is replaced with the entry ØØØ.

SEGMENT LIMIT is retained.

3 SPECIAL-NAMES.

Any existing entries referring to hardware names are deleted and flagged (D13); others are retained.

GETIME-IS-H6000-DATE, COLLATE-COMMERCIAL and COMPILE ERRORS entries are generated.

(b) INPUT-OUTPUT SECTION:

A reserved word test is accomplished on each SELECTED file name. If found, the rightmost digit is replaced with the character "Q" and flagged (D5).

1 FILE-CONTROL:

All SELECT statements are modified to acceptable H6000 formats. File codes are assigned as F1, F2,..... F9, FJ,.....FO.

The clause FOR CARDS/LISTING is appended to the appropriate files.

RANDOM ACCESS and ACTUAL KEY clauses are retained; all other clauses are deleted.

2 I-O-CONTROL:

The clause APPLY SYSTEM STANDARD FORMAT ON is generated for each file name SELECTed in the FILE-CONTROL paragraph.

NOTE: If stranger files are to be processed by the translated program, appropriate user modification must be made. (Ref H6000 COBOL Reference Manual, BS08.)

The SAME AREA clause is retained; all others are deleted.

(4) DATA DIVISION:

A reserved word test is accomplished for unique H6000 entries. If found, the rightmost digit is replaced with the character "Q" and flagged (D5).

(a) FILE SECTION:

Whenever an FD is found, the translator flags the FD (D6 and D29) due to a null VALUE OF ID clause and because FD clauses as input have been deleted and generates:

```
FD file-name

   LABEL RECORDS ARE STANDARD

   VALUE OF ID RECORDS ARE STANDARD
```

Whenever a SD is found, the translator flags the SD (D29) because SD clauses as input have been deleted and generates:

```
SD file-name
```

(b) WORKING-STORAGE (&CONSTANT) SECTION:

All COMPUTATIONAL, SYNCHRONIZED and JUSTIFIED clauses will be flagged (D8, D9 or D22). (Special attention should be given if these elements are imbedded in a record description.)

Two Ø1 levels (record descriptions) are generated to provide the H6000 data name TODAYS-DATE to the PROCEDURE DIVISION coding at object time.

ALL COPY....are flagged (D12) and retained in the source program as comments (* in column 7).

Word ALL is flagged (D19) because of restrictions on its use in H6000.

BZ is flagged (D2) as it is an unacceptable abbreviation of BLANK WHEN ZERO.

JS is flagged (D21) as it is an unacceptable abbreviation of JUSTIFIED.

MOD is deleted and flagged (D23).

OC is flagged (D24) as it is an unacceptable abbreviation of OCCURS.

PC is flagged (D25) as it is an unacceptable abbreviation of PICTURE.

SY is flagged (D26) as it is an unacceptable abbreviation of SYNCHRONIZED.

SZ is flagged (D27) as it is an unacceptable abbreviation of SIZE.

VA is flagged (D28) as it is an unacceptable abbreviation of VALUE.

If DATE-COMPILED is used as a value for an item, it is changed to Ø and flagged (D45).

(5) PROCEDURE DIVISION:

All DECLARATIVES (if present) are copied.

Code is generated to place the current date in the generated data name TODAYS-DATE.

A reserved word test is accomplished for unique H6000 entries. If found, the rightmost digit is replaced with the character "Q" and flagged (D5).

All SYMBOLIC coding is flagged (D4) and retained in the source program as comments (* in column 7).

ACCEPT, DISPLAY & STOP (LITERAL) verbs are flagged (D1, D2, D3) as requiring manual modification.

The IF verb coding is modified to include explicit AND/OR connectors between all non-numeric literals in statements with implied connectors; these connectors are not inserted between data names or numeric literals in these

statements. Also, in the operator "= TO", the word "TO" is not deleted. For these reasons the IF verb is always flagged D7. In the event that an IF statement contains a non-numeric literal greater than 50 characters, it is truncated to 50 characters and is flagged (D17).

READ and RETURN statements in H6000 COBOL require explicit AT End coding. All READ and RETURN verbs are flagged (D14 and D15) for programmer interrogation.

The ALTER verb is flagged (D18) as some installations restrict its use.

Word ALL is flagged (D19) because of restrictions on it use on H6000.

CLOSE statements flagged (D31) because, if used, PURGE or RELEASE are not acceptable. An attempt is made to delete the first occurrence of either PURGE or RELEASE.

ALL COPY ... are flagged (D12) and retained as comments (* in cc 7).

FILL ... is flagged (D32) and retained as comments (*in cc 7).

If GO is not followed by the word TO, it is flagged (D34).

INPUT-OUTPUT is changed to I-O and flagged (D41).

MOD is deleted and flagged (D23).

O-I is changed to I-O and flagged (D35).

ALL OPEN statements are flagged (D36) as the words ACCESS, LOCK or REVERSED are not acceptable and if used, should be removed by the programmer.

ALL SORT statements are flagged (D10) as the words END, ERROR, LOCK, PURGE, RELEASE and RUN are not acceptable and if used, should be removed by the programmer.

SW1 thru SW8 are flagged (D37) as these are B3500 switches.

TRACE is flagged (D38) and retained as comment (*in column 7).

UNLOCK ... is deleted and flagged (D39).

WAIT ... is deleted and flagged (D34).

WRITE statements are flagged (D30) as format differs between B3500 and H6000.

ZIP ... is flagged (D11) and retained as comment.

KEY CONVERSION and LOCK clauses are removed from SEEK statements and flagged (D42).

Any reference to UNDIGIT LITERALS is flagged (D43).

DATE-NAMES followed by subscript without an intervening space are flagged (D44).

e. Known coding differences that are not modified or flagged by the translator:

(1) ENVIRONMENT DIVISION:

The COPY is not flagged.

(2) DATA DIVISION:

Maximum size for non-numeric literals is 132 characters.

Maximum size for numeric literals is 18 characters.

(3) PROCEDURE DIVISION:

The statement ADD ... TO ... GIVING ..., the word TO is illegal.

For printed reports, existing maximum vertical line counts must be reduced if WWMCCS Security Header and Trailer Caveats are utilized.

ROUNDED may not be used on remainder in DIVIDE statements.

f. A side-by-side listing of the B3500 program and the translated program is produced.

g. Diagnostic codes are placed on the side-by-side listing. A list of the codes and their meaning is given in Figure 20-02.

h. A source program file position index list is produced immediately after the diagnostic codes.

20.1.1.2 References:

a. AFM 171-121.

b. B3500 Information Processing Systems - COBOL.

c. H6000 COBOL Reference Manual, BS∅8.

d. H6000 COBOL Users Guide, BS∅9.

e. H6000 Control Cards, BS19.

f. H6000 Utility, BQ66

g. H6000 File System, BR38.

h. H6000 GCOS, BR43.

i. AFM 171-604, Vol I.

20.1.2 Staff Input Requirements. Input will consist of COBOL source programs on card, tape or disk. One parameter card is required for each program to be translated. (Ref Figure 20-01.)

20.1.2.1 Input Formats. COBOL source program input will be a card deck or a tape or disk file containing card images. A tape or disk file will be in system standard format.

One parameter card will be input for each program to be translated. The format is described by Figure 20-01.

20.1.2.2 Composition Rules. A maximum of 50 programs may be input in one activity.

20.1.2.3 Input Vocabulary. N/A.

20.1.2.4 Sample Inputs:

a. SAMPLE JCL FOR CARD INPUT (SOURCE). If the source deck contains $\binom{11}{\emptyset}$ and/or $\binom{12}{\emptyset}$ punches, program ZAAØFO should be executed as the first activity in the job stream. (Ref AFM 171-604, Part Four, Volumes I and II.)

```
Col 1          Col 8           Col 16
    $              IDENT           (User Specified)
    $              USERID          (User Specified)
    $              LIBRARY         LA
    $              USE             ZABUFO
    $              ENTRY           C.ZABU
    $              EXECUTE
    $              LIMITS          n1,23K,,n2
    $              TAPE9           LA,XØD,,nnnnn,,USER-LIBRARY
    $              DATA            C2
                   [Input SP891-BU1
                    Parameter Cards (1-n)]  Figure 20-01.
    $              DATA            C1
                   Source Deck 1
                   .
                   .
                   .
                   Source Deck n
    $              TAPE9           F2   (output file description -
                                         to UTILITY or COBOL)
  or
    $              PRMFL           F2
    $              SYSOUT          P1
    $              ENDJOB
```

where n = a maximum of 50
n1 = number of parameter cards X .04 hrs
n2 = total number of source cards of all programs to be translated X 2.5 (SYS-OUT limits)

* b. SAMPLE JCL FOR TAPE OR DISK INPUT (SOURCE). Tape input to the translator must be converted via program ZAT1FO (Ref AFM 171-604, Part Five, Volumes I and II). B3500 MFSOLT tapes contain a directory plus one file for each source program. To get the correct number of programs converted (ZAT1FO), the user must add one to the number of files input (Ref Figure 10-02, INPUT FILE/REEL Field). ZABUFO will accept only the first B3500 source program output from ZAT1FO. If the remaining programs are to be translated, each program must be input individually to ZABUFO via UTILITY or input as cards.

```
Col 1          Col 8           Col 16
    $              IDENT           (User Specified)
    $              USERID          (User Specified)
    $              LIBRARY         LA
    $              USE             ZABUFO
    $              ENTRY           C.ZABU
    $              EXECUTE
    $              LIMITS          n1,23K,,n2
    $              TAPE9           LA,XØD,,nnnnn,,USER-LIBRARY
    $              DATA            C2
                   [Input SP891-BU1
                    Parameter Cards (1-n)]  Figure 20-01.
```

```
Col 1          Col 8          Col 16
    $          TAPE9              F1   (input file description -
                                        from ZAT1FO)
 or($          PRMFL              F1)
    $          TAPE9              F2   (output file description -
                                        to UTILITY or COBOL)
 or($          PRMFL              F2)
    $          SYSOUT             P1
    $          ENDJOB
```

where n, n1, and n2 are as defined in paragraph 20.1.2.4.a.

c. H6000 COBOL COMPILATION AFTER ZABUFO. A source program can be selected from the translator output file by UTILITY as input to compilation. The following example of JCL flow briefly describes this function.

```
Col 1          Col 8          Col 16
    $              IDENT          (User Specified)
    $              USERID         (User Specified)
    $              UTILITY
    $              TAPE9          F1   (input file description -
                                        from Translator)
 or($              PRMFL          F1)
    $              FILE           F3,A1S     (output file description -
                                               input to COBOL)
    $              FUTIL          F1,F3,RWD/F1/,(SKIP/n-1/),
    $              ETC            COPY/1F/,RWD/F1/
    $              COBOL          options
    $              UPDATE
    $              ALTER
                   .
                   .              (source program changes) optional
                   .
    $              FILE           S*,A1R     (COBOL source input -
                                               from UTILITY)
    $              ENDJOB
```

where n = source program position on Translator created file (reference Translator Index Listing). These activities could be placed immediately preceding the $ENDJOB card of the translator activity. ($ SNUMB, IDENT and USERID cards must be deleted and Logical Unit Designators (LUDs) must bind the files between activities.) If only one source program is translated, the translator output file can be used as input to COBOL (S*), bypassing the UTILITY activity. (A LUD must bind the files between activities.)

20.1.3 Output Requirements. Output will be a translated program on tape or disk. A side-by-side listing of the B3500 and H6000 program is generated with a flag legend. Additionally, a source program file position index is listed after the last source program flag legend.

20.1.3.1 Output Formats. Translated programs will be on tape or disk in system standard format.

20.1.3.2 Sample Outputs. N/A.

20.1.3.3 Output Vocabulary. N/A.

20.1.4 Utilization of System Outputs. N/A.

20.1.5 Recovery and Error Correction Procedures. N/A.

RECORD LAYOUT			
FILE TITLE Parameter Cards CZABUFOAU	RECORD TITLE Program Identifier Parameter Cards		CLASSIFICATION Unclassified RCD/PCN SP891-BU1
FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
PIC	01 - 11	AN	Enter "PCNSP891BU1" - first card only.
Filler	12-19		Blank.
B3500 PROGRAM-ID	20 - 25	AN	This entry must exactly match the first six characters of the input Program-ID. If less than six characters, left justify the entry.
H6000 PROGRAM-ID	26 - 31	AN	Program-ID to be placed in the Program-ID paragraph and in cc 73-78 of each output source card.
Filler	32		Blank.
SOURCE ORIGIN(S)	33 - 36	AL	Active only for the first parameter card of each translator activity. If "CARD", all source programs will be card input. If blank, all source programs are input from tape or disk under JCL control.
Filler	37 - 80		Blank.
NOTE: One parameter card is required per input source program. A maximum of 50 parameter cards is allowed per activity. Where multiple parameter cards are input, the product control number (PCN) will be reflected on the first card only.			

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 20-01. Program Identifier Parameter Cards (ZABUFO)

INDEX OF THE DIAGNOSTIC CODES ON THE SIDE-BY-SIDE

CODE DIAGNOSTIC COMMENT
D1 REFERENCE MANUAL FOR PROPER ACCEPT FORMAT
D2 REFERENCE MANUAL FOR PROPER DISPLAY FORMAT
D3 STOP (LITERAL) SHOULD BE USED WITH DISCRETION
D4 ENTER SYMBOLIC NOT ACCEPTABLE ON H6000. MADE COMMENTS
D5 DATA-NAME USED IS RESERVED WORD. LAST CHAR CHANGED TO Q
D6 VALUE OF ID TO BE INSERTED BY PROGRAMMER
D7 IF STATEMENT MAY REQUIRE FURTHER MODIFICATION
D8 CHECK COMP FIELDS FOR COMPATABILITY
D9 CHECK SYNC FIELDS FOR COMPATABILITY
D10 IN SORT-END,ERROR,LOCK,PURGE,RELEASE,RUN-NOT ACCEPTABLE
D11 ZIP NOT ACCEPTABLE ON H6000. MADE THIS CARD A COMMENT
D12 COPY REQUIRES A LIBRARY. MADE THIS CARD A COMMENT
D13 SPECIAL NAMES REFERRING TO HARDWARE-NAMES DELETED
D14 INSURE READ HAS EXPLICIT AT-END CLAUSE
D15 INSURE RETURN HAS EXPLICIT AT-END CLAUSE
D16 QUOTE WITHIN A QUOTE ILLEGAL, CHANGED TO COLON
D17 IN IF STATE; UNABLE TO PROCESS LITERAL GRTR THAN 50 CHARS
TRUNCATED IT. CHANGE TO ORIGINAL VALUE BEFORE COMPILATION
D18 SOME INSTALLATIONS RESTRICT THE USE OF THE ALTER VERB
D19 REFERENCE MANUAL FOR PROPER USE OF WORD ALL
D20 BZ SHOULD BE CHANGED TO BLANK ZERO
D21 JS SHOULD BE CHANGED TO JUST OR JUSTIFIED
D22 CHECK JUSTIFIED FIELDS FOR COMPATABILITY
D23 MOD HAS BEEN DELETED
D24 OC SHOULD BE CHANGED TO OCCURS
D25 PC SHOULD BE CHANGED TO PIC OR PICTURE
D26 SY SHOULD BE CHANGED TO SYNC OR SYNCHRONIZED
D27 SZ SHOULD BE CHANGED TO SIZE
D28 VA SHOULD BE CHANGED TO VALUE
D29 FD AND SD CLAUSES AS INPUT HAVE BEEN DELETED
D30 REFERENCE MANUAL FOR PROPER WRITE FORMAT
D31 CLOSE WITH PURGE OR RELEASE NOT ACCEPTABLE
IF USED, ATTEMPTED TO DELETE FIRST OCCURRENCE
D32 FILL NOT ACCEPTABLE ON H6000, MADE THIS CARD A COMMENT
D33 GO MUST BE FOLLOWED BY THE WORD TO
D34 WAIT NOT ACCEPTABLE ON H6000, DELETED IT
D35 O-I HAS BEEN CHANGED TO I-O
D36 IF USED, OPEN WITH ACCESS,LOCK OR REVERSED NOT ACCEPTABLE
D37 SW1 THRU SW8 NOT ACCEPTABLE ON H6000
D38 TRACE NOT ACCEPTABLE ON H6000, MADE THIS CARD A COMMENT
D39 UNLOCK NOT ACCEPTABLE ON H6000, DELETED IT
D40 WITH LOCK HAS BEEN REMOVED FROM READ STATEMENT
D41 INPUT-OUTPUT HAS BEEN CHANGED TO I-O
D42 KEY CONVERSION AND/OR LOCK DELETED FROM SEEK STATEMENT
D43 UNDIGIT LITERALS ARE UNACCEPTABLE ON H6000
D44 H6000 REQUIRES A SPACE BETWEEN DATA-NAME AND SUBSCRIPT
D45 DATE-COMPILED UNACCEPTABLE VALUE ON H6000;CHANGED TO ZERO

FIGURE 20-02. Index of Diagnostic Codes

PART ELEVEN - TAPE CERTIFICATION PROGRAM

SECTION 21. SYSTEM SUMMARY

21.1 System Application. The purpose of the Tape Certification Program is to provide a means of certifying magnetic tapes for those installations not possessing an electronic tape certifier. It provides the capability of locating and identifying defective sections of magnetic tape. This will enable defective and marginal tapes to be identified and removed from the tape library.

21.2 System Operation. N/A.

21.3 System Configuration. The tape certification program was written for use on the H6000 computer system with tape I/O devices.

21.4 System Organization. The Tape Certification Program consists of one (1) stand-alone program (ZAT3FO).

21.4.1 ZAT3FO - Tape Certification Program. ZAT3FO is a GMAP coded program using any 9-track tape as input.

21.5 Performance. N/A.

21.6 Data Base. N/A.

21.7 General Descriptions of Inputs, Processing, Outputs:

21.7.1 Inputs. Any 9-track tape the user wishes to be certified as to its physical condition.

21.7.2 Processing. The Tape Certification Program is a stand-alone, self-contained program. All checking, logging, and error recovery are internal to the program.

21.7.3 Outputs. N/A.

SECTION 22. STAFF FUNCTIONS RELATED to TECHNICAL OPERATIONS

22.1 ZAT3FO - Tape Certification Program. The function of this program is to certify the condition of magnetic tapes.

22.1.1 Initiation Procedures. The following JCL is used to execute ZAT3FO:

```
Col 1       Col 8          Col 16

   $           IDENT          (User Specified)
   $           USERID         (User Specified)
   $           LOWLOAD
   $           OPTION         NOSETU
   $           LIBRARY        LA
   $           USE            ZAT3FO
   $           ENTRY          ZAT3FO
   $           EXECUTE
   $           LIMITS         NNN,5K,,2000
   $           TAPE9          LA,XØD,,nnnnn,,USER-LIBRARY
   $           FILE           P*,NULL
   $           TAPE9          FA,X1D(TAPE TO BE CERTIFIED)
   $           ENDJOB
```

41

22.1.1.1 Program Notes:

a. NNN = $\frac{\text{Number of Reels}}{2}$. Approximately .005 hours are required to certify one reel of tape. The user should enter a value dependent on the number of tapes to be certified. If the number of reels is less than ten, use the default option.

22.1.1.2 References:

a. H6000 Equipment Operators Manual, DA33.

b. H6000 Control Card Reference Manual #BS19.

22.1.2 Staff Input Requirements. N/A.

22.1.2.1 Input Formats. N/A.

22.1.2.2 Composition Rules. N/A.

22.1.2.3 Input Vocabulary. N/A.

22.1.2.4 Sample Inputs. N/A.

22.1.3 Output Requirements. N/A.

22.1.3.1 Output Formats. N/A.

22.1.3.2 Sample Outputs. N/A.

22.1.3.3 Output Vocabulary. N/A.

22.1.4 Utilization of System Outputs.

a. The following information messages are provided on the execution report under the activity identifier; e.g., *ACTY-01 $ CARD 0005 GELOAD:

(1) TAPE REEL NUMBER IS NNNN. This message identifies the tape being certified. NNNN is the reel number.

(2) UNABLE TO PERFORM I/O. This message results when an irrecoverable I/O error is encountered. This situation is sometimes the result of hardware malfunction, a tape not having an end of tape (EOT) marker, or the tape unit not sensing the EOT marker.

(3) NO TAPE ERRORS DETECTED. Indicates that the tape was not found to be defective.

(4) TAPE DEFECTIVE BOT PLUS NNNN FEET. A defective section of tape was encountered NNNN (number of feet) feet from beginning of tape (BOT). This message will occur for each error detected.

b. ZAT3FO will allocate a tape handler and the operator should ready the unit with the first tape to be certified. At the end of the certification process for each tape, the following message will appear on the console:

***TAPE CERTIFIER S#XXXXX, (T)ERM OR (C)ONTINUE

If there is another tape to be certified, the operator should type a "C". A "T" will terminate the run. If "C" is entered, a standard mount message will be issued for the tape handler.

c. The following message will appear if no label record exists on the tape:

**ENTER REEL NUMBER i-cc-uu

The reel number should be entered.

22.1.5 Recovery and Error Correction Procedures. N/A.

PART TWELVE - SYSTEM STARTUP CREATE/UPDATE

SECTION 23. SYSTEM SUMMARY

23.1 System Application. The purpose of this system is to create and maintain a tape which contains a copy of the current boot-deck. This system enables an initial tape to be created from card input. It also provides the ability to update this boot-deck tape with any section(s) of the boot-deck by building a new tape with the new section(s) applied in the proper sequence.

23.2 System Operation. N/A.

23.3 System Configuration. The system startup boot-tape program was written for use on the H6000 computer system with card reader and tape I/O devices.

23.4 System Organization. This system consists of one (1) stand-alone program (ZAT4FO).

23.4.1 ZAT4FO - H6000 System Startup Tape Create/Update Program. This program provides the capability to create/update and print the H6000 system startup boot-tape.

23.5 Performance. N/A.

23.6 Data Base. N/A.

23.7 General Description of Inputs, Processing, Outputs. See specific information of inputs, processing, and outputs in Section 24.

SECTION 24. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

24.1 ZAT4FO - H6000 System Startup Tape Create/Update Program. The function of this program is to create/update and print the H6000 system startup boot-tape. ZAT4FO accepts as input any section(s) of the boot-deck and the current boot-tape and creates a new boot-tape.

24.1.1 Initiation Procedures. To initiate execution of ZAT4FO, the user may execute the Job Control Language (JCL) in paragraph 24.1.2.4.

24.1.1.1 Program Notes:

a. BMC is used to process the input and output since they are not standard format (mixed BCD and binary).

b. The boot-deck sections to be updated must be input in the same order as they appear on the current boot-tape (See paragraph 24.1.2.4.c).

c. ZAT4FO is designed to read 7 or 9 track tapes.

d. Input cards, object cards, and the sections on tape are sequence checked. If they are out of sequence, an error will be printed and the activity will terminate with an A1 abort.

24.1.1.2 References:

a. H6000 Startup and Operations, DD33.

b. H6000 Control Cards, BS19.

c. H6000 Bulk Media Conversion, BP30.

24.1.2 Staff Input Requirements. Input is the portion(s) of the boot-deck which need updating and a current boot-tape. The boot-deck sections to be updated will be preceded by a Product Control Card (PCN). See Figure 24-01. User requirements will determine frequency of processing.

24.1.2.1 Input Formats:

a. Reference H6000 Startup and Operation Manual, DD33 for description of boot-deck sections to be updated.

b. $BOOT-D Card Format. See Figure 24-01.

c. $PRINT Card Format. See Figure 24-01.

d. Product Control Card (PCN) Format. See Figure 24-01.

24.1.2.2 Composition Rules. The boot-deck sections to be updated must be input in the same sequence as they appear on the current boot-deck tape. (See paragraph 24.1.2.4.c). These boot-deck sections must always be preceded by a "PCN" card and for the initial creation a "$BOOT-D" card is required. (See Figure 24-01).

24.1.2.3 Input Vocabulary. N/A.

24.1.2.4 Sample Inputs:

a. Job Control Language (JCL) for the initial creation of H6000 system startup boot-tape:

```
Col 1      Col 8        Col 16

    $          IDENT        (User Specified)
    $          USERID       (User Specified)
    $          CONVER
    $          READ         IN,A1R
    $          INPUT        MBIN
    $          TAPEn        OT,X1D,,,,BOOTAPE
    $          OUTPUT       B27,MIXL,NLABEL,NSER
    $          ENDJOB
```

(where "n" on the $ TAPE card is a 7 or 9 depending on the type of drive to be used for the tape bootload).

b. Job Control Language (JCL) for H6000 system startup tape update.

```
Col 1      Col 8        Col 16

    $          IDENT        (User Specified)
    $          USERID       (User Specified)
    $          CONVER       NSPIN
    $          INPUT        MBIN
    $          READ         IN
    $          FILE         OT,X1S,5ØL
    $          LIBRARY      LA
    $          USE          ZAT4FO
    $          ENTRY        C.ZAT4
    $          EXECUTE
    $          LIMITS       1Ø,15K
    $          TAPE9        LA,XØD,,nnnnn,,USER-LIBRARY
    $          TAPE9        F1,B1D,,(I/P Reel #),,(File ID)
    $          SYSOUT       P2
    $          FILE         W3,X1R
    $          TAPE9        F4,B2D,,,,(O/P File ID)
    $          ENDJOB
```

c. When the BMC requests the card readers, input the bootdeck section(s) in sequence that are to be updated. For a list of the boot-tape, input the "$PRINT " control card followed by three (3) ***EOF cards. The boot-deck sections are:

(1) BOOT-DECK, ***EOF (precede with $BOOT-D Card)

(2) INIT-DECK

(3) $CONFIG, ***EOF

(4) $INITIALIZE, ***EOF

(5) $EDIT, ***EOF

(6) $FILES, ***EOF

(7) $PATCH, ***EOF

(8) $LOAD, ***EOF

(9) ***EOF, ***EOF - MANDATORY

NOTE: See Figure 24-01 for Control Card Format.

24.1.3 Output Requirements. Output will be a new or updated boot-deck tape along with an update listing on the Printer.

24.1.3.1 Output Formats. N/A.

24.1.3.2 Sample Outputs. N/A.

24.1.3.3 Output Vocabulary. N/A.

24.1.4 Utilization of System Outputs. N/A.

24.1.5 Recovery and Error Correction Procedures. N/A.

RECORD LAYOUT

FILE TITLE	RECORD TITLE		CLASSIFICATION
Control Cards CZAT4FOAU	ZAT4FO Control Cards		Unclassified
			RCS/PCN SP891-T41
FILE DESCRIPTION	RECORD POSITIONS	TYPE/CLASS	SPECIAL INSTRUCTIONS
Product Control Card (PCN)			
C-PCN	01-11	AN	Enter "PCNSP891T41".
Filler	12-80		Blank.
Boot-Deck Control Card (Initial Creation)			
C-IDENT	01-07	AN	Enter "$BOOT-D".
Filler	08-80		Blank.
Print Only Control Card			
C-IDENT	01-07	AN	Enter "$PRINT ".
Filler	08-80		Blank.

NOTE 1: The "PCN" card is required for all runs.

NOTE 2: The "$BOOT-D" card is required only for initial creation of a new boot tape.

NOTE 3: The "$PRINT " card is required for a listing of the boot tape (no update).

NOTE 4: The "INIT" section does not require a control card for updating (just input the INIT deck intact). All other Update actions require the respective "$ " in cc 1-7 of the control card.

AF FORM 1190 AUG 70 PREVIOUS EDITIONS ARE OBSOLETE.

FIGURE 24-01. Record Layout for ZAT4FO Control Cards

* PART THIRTEEN - TAPE UNIT COMPATIBILITY TEST PROGRAM

SECTION 25. SYSTEM SUMMARY

25.1 System Application. The purpose of the Tape Unit Compatibility Test Program is to test the compatibility of the Magnetic Tape Handlers on H6000 computers.

25.2 System Operation. NA.

25.3 System Configuration. The Tape Unit Compatibility Test Program was written for use on the H6000 computer system with the System Console and Magnetic Tape Unit I/O devices.

25.4 System Organization. The Tape Unit Compatibility Test Program consists of one stand-alone program (ZAJ1FO).

25.4.1 ZAJ1FO - Tape Unit Compatibility Test Program. ZAJ1FO creates a Standard System Test Tape containing a string of random numbers written by each Tape Handler tested. Each Tape Handler being tested then attempts to read the random string written by each Tape Handler and prints a report of any read/write errors encountered during the test.

25.5 Performance. NA.

25.6 Data Base. NA.

25.7 General Description of Inputs, Processing, Outputs:

25.7.1 Inputs. Any 9-track scratch tape.

25.7.2 Processing. ZAJ1FO provides the capability to test specific Tape Handlers against one another to determine their compatibility.

25.7.3 Outputs. ZAJ1FO outputs a printed report of any read/write errors encountered during the test.

SECTION 26. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

26.1 ZAJ1FO - Tape Unit Compatibility Test Program. The function of this program is to test the compatibility of Magnetic Tape Handlers on H6000 computers.

26.1.1 Initiation Procedures. To initiate execution of ZAJ1FO, the user may execute the following JCL:

```
Col 1  Col 8   Col 16

   $      IDENT    (User Specified)
   $      USERID   (User Specified)
   $      LIBRARY  LA
   $      USE      ZAJ1FO
   $      ENTRY    ZAJ1FO
   $      EXECUTE
   $      LIMITS   ,5K
   $      TAPE9    LA,X1D,,nnnnn,,User Library
   $      ENDJOB
```

26.1.1.1 Program Notes. Library Tape LA is in sequential R* format.

26.1.1.2 References. H6000 Control Card Manual #BS19.

26.1.2 Staff Input Requirements. User requirements will determine frequency of processing.

26.1.2.1 Input Formats. NA.

26.1.2.2 Composition Rules. NA.

26.1.2.3 Input Vocabulary. NA.

26.1.2.4 Sample Inputs. NA.

26.1.3 Output Requirements. Output will be a read/write error report listing. User requirements determine the frequency of processing and disposition.

26.1.3.1 Output Formats. NA.

26.1.3.2 Sample Outputs. Reference figure 26-01.

26.1.3.3 Output Vocabulary. NA.

26.1.4 Utilization of System Output. The following information messages are output on ZAJ1FO - Tape Unit Compatibility Test Results, PCN: SP891-J11, File-ID: PZAJ1FO1U.

a. READ ON UNIT MMMMMM WRITTEN ON UNIT NNNNNN

"MMMMMM" and "NNNNNN" are six-digit numbers indicating a specified tape handler. Whenever this message appears it indicates a data discrepancy between the reading and writing tape handlers, possibly indicating that one of the two units is becoming incompatible. The appropriate personnel should be notified.

b. MMMMMM READ ERRORS FOR UNIT NNNNNN

"MMMMMM" indicates the total number of discrepancies encountered on the tape handler indicated by "NNNNNN." This will appear once for each tape handler tested. May be used as a diagnostic aid in determining which tape handler is becoming incompatible.

26.1.5 Recovery and Error Correction Procedures. If program aborts, rerun ZAJ1FO paying particular attention to giving correct responses on the system console to program initiated questions.

PREPARED 78 MAR 01 ZAJ1FO - TAPE UNIT COMPATIBILITY TEST RESULTS PCN SP891-J11

000000 READ ERRORS FOR UNIT 1

000000 READ ERRORS FOR UNIT 2

READ ON UNIT 3 WAS WRITTEN ON UNIT 1

000001 READ ERRORS FOR UNIT 3

READ ON UNIT 4 WAS WRITTEN ON UNIT 6

READ ON UNIT 4 WAS WRITTEN ON UNIT 6

000002 READ ERRORS ON UNIT 4

000000 READ ERRORS ON UNIT 5

000000 READ ERRORS ON UNIT 6

FIGURE 26-01. ZAJ1FO Sample Output

* PART FOURTEEN - STANDARD A1 ABORT SUBROUTINE

SECTION 27. SYSTEM SUMMARY

27.1 System Application. The purpose of the Standard A1 Abort Subroutine is to provide H6000 programmers a common subroutine that will eliminate redundant programming efforts. This subroutine is designed to cause a display of the user's A1 Abort code when there is a need to discontinue execution of a COBOL program or job stream caused by a predetermined error condition that would negate continued processing without being corrected.

27.2 System Operation. N/A.

27.3 System Configuration. The Standard A1 Abort Subroutine was written for use on the basic H6000 computer system.

27.4 System Organization. The Standard A1 Abort Subroutine is a callable object code subroutine that is called by a user program.

27.4.1 ZAS1FO - Standard A1 Abort Subroutine. ZAS1FO is designed to cause a display of the user's A1 Abort code when there is a need to discontinue execution of a COBOL program or job stream caused by a pre-determined error condition that would negate continued processing without being corrected.

27.5 Performance. N/A.

27.6 Data Base. N/A.

27.7 General Description of Inputs, Processing, Outputs:

27.7.1 Inputs. N/A.

27.7.2 Processing. The Standard A1 Abort Subroutine provides the H6000 COBOL programmer with the capability to call in a standard subroutine for user's A1 aborts that will eliminate redundant programming effort.

27.7.3 Outputs:

27.7.3.1 ZAS1FO. The user's A1 Abort code is output on the activity termination line of the execution report (P* file).

SECTION 28. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

28.1 ZAS1FO - Standard A1 Abort Subroutine. The function of this subroutine is to cause the display of the User's A1 Abort code on the activity termination line of the job execution report.

28.1.1 Initiation Procedures. The subroutine ZAS1FO must be available to the general loader on an object library to be loaded with the calling program. The object library may be made available to the loader by using the file code *L on a file control card which allocates the library or by using a $LIBRARY control card.

28.1.1.1 Program Notes:

a. The following is a coding example for calling ZAS1FO in a COBOL program:

```
$COBOL
 IDENTIFICATION DIVISION.

 WORKING STORAGE SECTION.

 PROCEDURE DIVISION.

 DISPLAY "ERROR MESSAGE."
 CALL ZAS1FO.

 STOP RUN.
```

28.1.1.2 References:

a. H6000 General Loader Manual, DD10.

b. H6000 COBOL Reference Manual, DD25.

c. H6000 GMAP Manual, DD08.

d. H6000 Control Card Reference Manual, DD31.

28.1.2 Staff Input Requirements. Number of User abort terminations will determine number and frequency of use for this subroutine.

28.1.2.1 Input Formats. N/A.

28.1.2.2 Composition Rules. N/A.

28.1.2.3 Input Vocabulary. N/A.

28.1.2.4 Sample Inputs. The following job control statements are required with the user's object library:

```
Col 1          Col 8          Col 16
    $              IDENT          (USER SPECIFIED)
    $              USERID         (USER SPECIFIED)
    $              LIBRARY        LA
    $              USE            ZAS1FO
    $              ENTRY          ZAS1FO
    $              EXECUTE
    $              LIMITS
    $              TAPE9          LA,XØD,,nnnnn,,User Library (with ZAS1FO)
    $              ENDJOB
```

28.1.3 Output Requirements. N/A.

28.1.3.1 Output Formats. N/A.

28.1.3.2 Sample Outputs. Reference figure 28-01.

28.1.3.3 Output Vocabulary. N/A.

28.1.4 Utilization of System Outputs. N/A.

28.1.5 Recovery and Error Correction Procedures. N/A.

```
*ACTY-03  SCARD  #0019  GELOAD  05/29/80  SW=400000000000
 *USERS A1 MME GEBORT  AT 037575 I=1060  SW=400000000000

 *WRAPUP BEGUN
 *NORMAL TERMINATION  AT 037104  I=5020 SW=000000000000

    START 15.294     LINES     730     PROC  0.0005     I/O  0.001     IU 5  MEMORY  16K
    STOP  15.306     LIMIT    5000     LIMIT 0.0500     LIMIT          CU 5  M*T     1000
    SWAP   0.000
    LAPSE  0.012 FC D  TYPE     BUSY     IP/AT     FP/RT     IS/#C MS/#E     ADDRESS T#/PK#

                 B* R  D450 *      20         1         1        24    24     0-08-01
                 R* R  D181 *      58         0         0         1     1     0-12-04
                 P*    SYOUT
                 *L R  D450 P      36         0         0         2     2     0-08-02
                 L* R  S181 *    1275         0         0      1000  1000R    0-12-02
```

* PART FIFTEEN - FILE ACCESS INQUIRY SUBROUTINE

SECTION 29. SYSTEM SUMMARY

29.1 System Application. The purpose of the File Access Inquiry Subroutine is to provide application programs the following information about a cataloged file:

a. The date that the last activity which wrote to the file relinquished its allocation. (MMDDYY - one word BCD)

b. The time that the last activity which wrote to the file relinquished its allocation. (HHMMSS - one word BCD)

c. The current total number of allocations of any kind to the file since it was created. (999999 - one word BCD)

d. The date of the last allocation of any kind. (MMDDYY - one word BCD)

e. The date that the file was created. (MMDDYY - one word BCD)

29.2 System Operation. N/A.

29.3 System Configuration. The File Access Inquiry Subroutine will run on any H6000 configuration with GCOS.

29.4 System Organization. This system consists of a single GMAP subroutine (ZAS2FO) which is called by application programs.

29.4.1 ZAS2FO - File Access Inquiry Subroutine. The function of the File Access Inquiry Subroutine is to provide a limited interface between COBOL application programs and the File Management Supervisor.

29.5 Performance. N/A.

29.6 Data Base. N/A.

29.7 General Description of Inputs, Processing, Outputs:

29.7.1 Inputs. N/A.

29.7.2 Processing.

a. ZAS2FO has several entry points. This is to give the user the option of identifying the file to be monitored by COBOL file name, FMS catalog/file name or by activity file code. The entry points and their required first parameters are as follows:

(1) FINQ01 - COBOL file name or GMAP FCB location.

(2) FINQ02 - COBOL file name or GMAP FCB location.

(3) FINQ03 - FMS catalog/file name.

(4) FINQ04 - Activity file code.

b. Processing is basically the same for each entry point. First ZAS2FO verifys that there are from two to six parameters. For entry points FINQ01, FINQ02 and FINQ04 a file code is extracted from the first parameter to identify the the file to FMS. For entry point FINQ03 the catalog/file name is reformatted for a separate FMS request. After FMS returns a buffer of information about a file ZAS2FO extracts those fields which contain the requested information and passes the information back to the application program's storage area. Entry point FINQ02 differs from the others in that it also de-allocates the file and passes back the

current time and date as the time and date that the file was last de-allocated from a writer. The format and order of the second through sixth parameters is as given in paragraph 29.1.

29.7.3 Outputs. N/A.

SECTION 30. STAFF FUNCTIONS RELATED TO TECHNICAL OPERATIONS

30.1 ZAS2FO - File Access Inquiry Subroutine:

30.1.1 Initiation Procedures. Figure 30-1 shows examples of calls to ZAS2FO from a COBOL program.

30.1.1.1 Program Notes:

a. The actual file names and data names can be different than those shown in figure 30-1.

b. The number of parameters can vary but they must be in the order shown in figure 30-1.

c. No PRMFL control card is required for entry point FINQ03.

d. Catalog/file names passed to entry point FINQ03 must be left justified and must end with a blank.

e. Only 10 levels of cataloging are supported by entry point FINQ03.

f. Except for entry point FINQ02 all of the information passed back by ZAS2FO is from the FMS type 2 record for the file.

g. The date of the last allocation and the current total number of allocations of any kind will reflect the fact that the user of entry points FINQ01, FINQ02 and FINQ04 must have allocated the file record with a PRMFL control card.

h. The time and date of the last writing to a file will not be recorded in the FMS type 2 record until after the file has been de-allocated by FINQ02 or the activity termination.

i. FINQ02 will de-allocate the file and return the current date and time as the date and time that the file was last de-allocated from a writer.

30.1.1.2 References:

a. H6000 General Loader Manual, DD10.

b. H6000 COBOL Reference Manual, DD25.

c. H6000 GMAP Manual, DD08.

d. H6000 Control Card Manual, DD31.

e. H6000 File Management Supervisor Manual, DD45

30.1.2 Staff Input Requirements. N/A.

30.1.2.1 Input Formats. N/A.

30.1.2.2 Composition Rules. N/A.

30.1.2.3 Input Vocabulary. N/A.

30.1.2.4 **Sample Inputs**:

```
Col 1  Col 8  Col 16

      $       IDENT
      $       USERID
      $       OPTION  COBOL
      $       COBOL
                                (the program which calls ZAS2FO)
      $       LIBRARY LB
      $       EXECUTE
      $       TAPE9   LB,TID,,nnnnn,,file-id  (of the release tape)

      $       ENDJOB
```

30.1.3 **Output Requirements**. N/A.

30.1.3.1 **Output Formats**. N/A.

30.1.3.2 **Sample Outputs**. N/A.

30.1.3.3 **Output Vocabulary**. N/A

30.1.4 **Uilization of System Outputs**. N/A

30.1.5 **Recovery and Error Correction Procedures**. N/A

```
Col 1  Col 8  Col 16

    $       COBOL

            FILE CONTROL.
              SELECT FILE-NAME AND ASSIGN TO F1.
            FILE SECTION.
            FD  FILE-NAME ...

            WORKING-STORAGE SECTION.

            77  FILE-CODE PIC  X(6) VALUE "0000F2".
            77  CAT-FILE  PIC X(16) VALUE "catalog/file3".
            77  W-DATE    PIC  X(6).
            77  W-TIME    PIC  X(6).
            77  N-ACS     PIC  9(6).
            77  A-DATE    PIC  X(6).
            77  C-DATE    PIC  X(6).

            PROCEDURE DIVISION.
            CALL FINQ01 USING FILE-NAME, W-DATE, W-TIME.
            CALL FINQ02 USING FILE-NAME, W-DATE, W-TIME.
            CALL FINQ03 USING CAT-FILE, W-DATE, W-TIME.
            CALL FINQ04 USING FILE-CODE, W-DATE, W-TIME.

                (OR FOR MORE INFO ABOUT A FILE)

            CALL FINQ01 USING FILE-NAME, W-DATE, W-TIME, N-ACS.
            CALL FINQ01 USING FILE-NAME, W-DATE, W-TIME, N-ACS, A-DATE.
            CALL FINQ01 USING FILE-NAME, W-DATE, W-TIME, N-ACS, A-DATE, C-DATE.

                (AND SO-ON FOR FINQ02, 01,04)
```

FIGURE 30-01. CAS2FO Example of Calls

BY ORDER OF THE SECRETARY OF THE AIR FORCE

OFFICIAL

DAVID C. JONES, General, USAF
Chief of Staff

JAMES J. SHEPARD, Colonel, USAF
Director of Administration

SUMMARY OF REVISED, DELETED, OR ADDED MATERIAL

This revision completely redocuments the P891 system IAW revised AFM 171-100 standards dated 1 November 1975. The following documents are deleted as of 1 December 1976:

H6000 User Advisory #20, dated 26 Aug 74.
H6000 User Advisory #22, dated 24 Oct 74.
H6000 User Advisory #23, dated 8 Mar 74.
H6000 User Advisory #25, dated 8 Apr 74.
H6000 User Advisory #26, dated 8 Apr 74.
H6000 User Advisory #27, dated 3 Jun 74.
H6000 User Advisory #30, dated 6 Jan 75.
H6000 User Advisory #31, dated 26 Aug 74.
H6000 User Advisory #34, dated 18 Sep 74.
H6000 User Advisory #35, dated 24 Oct 74.
H6000 User Advisory #36, dated 24 Oct 74.
H6000 User Advisory #37, dated 4 Dec 74.
H6000 User Advisory #39, dated 6 Jan 76.
H6000 User Advisory #40, dated 17 Jan 75.
H6000 User Advisory #44, dated 31 Mar 75.
H6000 User Advisory #45, dated 25 Aug 75.
H6000 User Advisory #46, dated 17 Oct 75.
H6000 User Advisory #47, dated 17 Oct 75.
Gunter H6000 Software Advisory Notice, SDT014, dated 25 Sep 73.
Gunter H6000 Software Advisory Notice, SDT016, dated 17 Mar 74.
Gunter H6000 Software Advisory Notice, SDT045, dated 12 Feb 75.
Gunter H6000 Software Advisory Notice, SDT055, dated 15 Aug 75.
Gunter H6000 Software Advisory Notice, SDT057, dated 28 Oct 75.
Gunter H6000 Software Advisory Notice, SDM064, dated 18 Feb 76.