

QC
874.3
.U61
no.4
c.2

National Weather Service, Central Region
Computer Programs and Problems
NWS CRCP - No. 4



PROGRAMS SAVALL AND STORALL

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, KS

July 1982

noaa

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION /

National Weather
Service



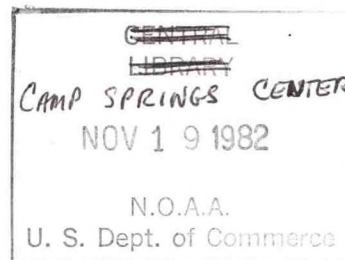
QC
874.3
461
no. 4
C.2

NOAA, National Weather Service, Central Region
Computer Programs and Problems

NWS CRCP - No. 4

PROGRAMS SAVALL AND STORALL

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, KS



July 1982

UNITED STATES
DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Oceanic and
Atmospheric Administration
John V. Byrne, Administrator

National Weather
Service
Richard E. Hallgren, Director



32 02951

PROGRAMS SAVALL AND STORALL

Warren E. Sunkel
National Weather Service Forecast Office
Topeka, Kansas

I. GENERAL INFORMATION

A. Introduction

SAVALL and STORALL are functionally related background programs which perform tasks similar to the AFOS SAVE and STORE commands. The difference from the foreground commands is that these programs operate on an entire category of AFOS database products rather than on a single product. Program SAVALL moves all products of a given category from the AFOS database to a disk directory. Conversely, program STORALL moves all files of a given category from a disk directory to the AFOS database.

Database products of a semi-permanent nature, such as procedures (PCD's), message composition preformats (MCP's), and map backgrounds, can be easily saved as backup files on a floppy disk. In the event of a database failure or a new software load, the backup files can be quickly moved to the newly created database. SAVALL and STORALL can also be used to update the date/time groups of these semi-permanent files at the beginning of the year. Because the AFOS software stores products by creation time in Julian minutes, a date/time group near the first of January will ensure that an updated version of a given product will always be stored as the latest version.

B. Program Environment

Both programs are designed to operate in the background partition of the AFOS minicomputer at either a WSFO or a WSO. The programs are written in Data General's assembly language for Eclipse computers. Input/output is performed by system calls to the Real Time Disk Operating System (RDOS). Linkage and initialization routines are supplied by the FORTRAN Library (FORT.LB).

Either program will operate in a region of approximately 4K of core memory.

The only disk files created are those products saved by program SAVALL. Normally these files will be on a floppy disk, so the only space required on the cartridge disks is for the program save files (-.SV) themselves. If it is more convenient, or if cartridge disk space is limited, the save files may be kept on the same floppy with the saved products.

Program execution time is dependent on the I/O device used for the saved products and on the number of products moved. Approximate times for a floppy disk and medium length AFOS products (2 to 3 screen pages) are 5 seconds to save a product and 10 seconds to store a file.

Each program requires three I/O channels. SAVALL requires one for input from the operator, one for disk output, and one to read the local SKEL. STORALL requires one for operator input, one for disk input and output, and one to read SYS.DR from the input device. All channels are fetched with the .GCHN system call.

C. References

Routines for retrieving and storing database products are described in AFOS Programming Note 90, "Background Programming with AFOS," by Mike Schuster.

Program STORALL uses the RDOS system directory (SYS.DR) to determine the filenames of all disk resident files. The format of SYS.DR is described in the Data General publication "Real Time Disk Operating System Reference Manual," number 093-000129, dated 1975.

II. APPLICATION PROGRAM DESCRIPTION

A. Program Structure and Description: SAVALL

SAVALL begins with an interactive session with the operator at the printer/keyboard (Dasher). The program first asks for the name of the destination directory. An attempt is made to initialize this directory. Processing will continue only if the system call completes successfully or if the directory is already initialized. Any other error, such as "directory does not exist," will cause a fatal error.

The program then asks if existing disk files are to be overwritten.

Operator instructions are typed, and the main loop is entered. An asterisk prompt (*) is printed each time the loop is executed. At this point the operator may specify the NNN of the AFOS products to be saved or may hit RETURN to exit the loop and terminate the program.

Upon receiving the NNN, the program creates the null key (CCCNNOOO), reading the local CCC from the SKEL if necessary. A key search is then initiated for the null key.

A second loop is entered to fetch the next key from DATAKEYO and move the current product to the specified destination directory. This loop continues until the CCCNNN of the next key does not match the CCCNNN of the current category, at which time control returns to the asterisk prompt.

If at least one version of the current product exists, the latest version is moved to disk using the following steps:

1. A filename is created using CCCNNNXXX and no extension. Trailing spaces in the XXX subfield are ignored.
2. Nine bytes of product key are written to disk followed by 9 spaces.
3. The first 23 bytes of the AFOS block 0 are skipped, and the remaining bytes are written to disk. If an end of text byte (203 octal) is encountered, it is the last byte output.
4. If additional AFOS blocks exist, the first 4 header bytes are skipped, and the remaining bytes are written to disk. If an end of text byte is encountered, it is the last byte output.
5. The product key and date/time group are verified on the console output.
6. The next key is fetched.

If no versions of the current product exist, a message specifying such is output to the console, and the next key is fetched.

The following subroutines are used to access AFOS database products. These subroutines reside on BG.LB and AFOSE.LB and have been duplicated on TOP.LB.

KSRCF - Search for a specific product key in DATAKEYO
NXKF - Return next key in DATAKEYO
RDBKF - Read a specific AFOS block for the current product

The following subroutines are used from TOP.LB:

KFILL - Expand a partial AFOS key into a full 9 characters
ADTG - Convert an AFOS Julian DTG to
month/day/hours/minutes
DTG - Convert a binary DTG to ASCII in the form MM/DD/YY
HH:MM

The linkage and initialization routines .FCAL and .I are used from FORT.LB.

SAVALL.SV is a self-contained executable file. No overlays are used. The load line is:

RLDR SAVALL TOP.LB FORT.LB

B. Program Structure and Description: STORALL

STORALL begins with an interactive session with the operator at the printer/keyboard (Dasher). The program first asks for the name of the source directory. An attempt is made to initialize this directory. Processing will continue only if the system call completes successfully or if the directory is already initialized. Any other error, such as "directory does not exist," will cause a fatal error.

Operator instructions are typed, and the main loop is entered. An asterisk prompt (*) is printed each time the loop is executed. At this point the operator may specify the NNN of the disk files to be stored or hit RETURN to exit the loop and terminate the program. If all valid files are to be stored, the operator should type the word ALL.

Upon receiving the filename template, the program reads SYS.DR on the source directory. SYS.DR is searched sequentially for valid filenames which match the template. A valid filename contains 7 to 9

characters and has no extension.

When a valid filename is encountered, that file is opened, and the 18-byte header is read. The CCCNNNXXX field of the header must match the filename, or the file will be skipped, and the search through SYS.DR will continue. If the product key matches the filename, the header is updated with the following values:

ADDRESSEE	000
DATE/TIME	Current System Date/Time
PRIORITY	5
TYPE	0

The header is written back to the disk file, and the channel is closed.

The disk file is moved to the AFOS database using subroutine USTOR. This subroutine calls subroutine FSTOR (from BG.LB) to initiate the storage process and then examines the file use count of the file to be stored. It waits for the file to be opened and closed by the foreground routines before returning to the calling program. Thus, the main program can move to the next file knowing that the current file has been completely read and is no longer needed.

Each file stored is verified on the console output. When SYS.DR has been completely searched, control returns to the asterisk prompt.

The following subroutines are used in program STORALL:

USTOR - Stores a disk file into the AFOS database and waits for completion before returning to the calling program (calls FSTOR)

KFILL - TOP.LB routine to expand an incomplete AFOS key to a full 9 characters

.FCAL - Linkage routine from FORT.LB

.I - Initialization routine from FORT.LB

STORALL.SV is a self-contained executable program. No overlays are used. The load line is:

RLDR STORALL USTOR TOP.LB FORT.LB

C. Special Note Concerning Graphics Products

All AFOS graphics products are assigned a CCCNNN of NMCGRP. The first character of the XXX subfield specifies the product category, such as B for map backgrounds, 5 for 500 millibar charts, etc.

When working with graphics products, both SAVALL and STORALL replace the CCCNNN with NMCGRP. The first character of the XXX subfield is then used as the template for locating matching products and files. For example: If the operator specifies the number 5, all products of the form NMCGRP5xx are saved or stored.

III. PROGRAM TESTING AND IMPLEMENTATION

A. Procedures for Program Implementation

Programs SAVALL and STORALL may reside on the main cartridge disks or on the individual floppies used to save database products. In either case, be sure to keep a backup copy of the .SV files on a floppy with other important applications programs.

At a WSFD, SAVALL.SV and STORALL.SV should reside on DPOF. Links should be created on DPO in the following manner:

```
DIR DPO
LINK SAVALL.SV DPOF:SAVALL.SV
LINK STORALL.SV DPOF:STORALL.SV
```

At a WSO, or where cartridge disk space is limited, move the programs to the floppy which is being used for saved products. This transfer can be done in one of two ways: (a) by moving the files from a floppy to DPO and then back to another floppy, or (b) by releasing the archive floppy and moving the files directly to another floppy.

In the following examples, actions to be taken are indicated in parentheses. Commands to be entered are indicated by CAPITAL LETTERS. All commands are to be entered at the printer/keyboard (Dasher) unless an ADM is specified.

Method A:

```
(Insert floppy containing program files in DP3)
DIR DP3
MOVE/V DPO SAVALL.SV STORALL.SV
RELEASE DP3
(Insert the floppy for saved products in DP3)
MOVE/V DP3 SAVALL.SV STORALL.SV
DELETE/V SAVALL.SV STORALL.SV
```

Method B:

```
(At an ADM, type RELEASE:DP2)
(Insert floppy containing program files in DP3)
DIR DP3
(Insert the floppy for saved products in DP2)
MOVE/V DP2 SAVALL.SV STORALL.SV
RELEASE DP3
(Replace the archive floppy in DP2)
(At an ADM type ARC:UP DP2)
```

Program SAVALL uses AFDS database products for input and creates an output file for each of the products saved. Program STORALL reads the saved files and stores them in the database. Thus, no additional input or output files are required.

Because both of the programs require an interactive session with the operator, they cannot be executed completely from an indirect or macro file.

B. Sample Run for Testing

Select a category of AFOS products and examine its PIL from an ADM. Enter P:NNN for local products, or P:CCCNNN for products from a remote node. For test purposes it is best to select a category with five to ten products. Most of the products should have valid date/time groups, but a few may be indicated by PURGED.

RELEASE:DP3 at an ADM and put a spare floppy in DP3. At the Dasher, open the directory containing the program .SV files: e.g., DIR DP3. If the files or links exist on DPO, this step is not necessary. Then type SAVALL.

Answer the questions asked by the program, and watch the products being moved to the disk. When all products in the specified category have been moved, an asterisk prompt will be typed. To exit the program, merely hit RETURN. NOTE: Even though the products listed have been moved to the floppy disk, they still exist in the AFOS database. Although it is not necessary to move them back to the database at this time, we shall do so in our next example.

=====

CAUTION: Moving products back to the database will cause the date/time group of each product to be set to the current system time. If you do not wish to change the DTG of any of the products saved, do not continue with the following example.

=====

After an R prompt from the CLI, type STORALL. Enter the source directory name used in the above example, i.e. the disk containing the saved products. When the asterisk prompt appears, type ALL. All valid files on the floppy disk will be stored back into the database and verified on the Dasher. To exit the program, hit RETURN after an asterisk prompt.

A sample interactive session is presented in Appendix 1.

When updating date/time groups at the beginning of a year, move all applicable products to a disk using program SAVALL. Before using STORALL to replace the products, purge the current database versions by entering the following commands from an ADM. For alphanumeric products, enter PURGE:CCCNNN to purge all products of a given category. (Note: The CCC subfield must be included in this command even if it is the local node.) For graphics products, enter PURGE:NMC GPHxxx for each product. Graphics must be purged individually.

The output from program SAVALL is exactly the same as that from the foreground SAVE command except that the last RDOS block does

not contain extraneous characters. The last byte of a SAVALL file is the end of text marker (203 octal). An FPRINT of a sample SAVALL file is shown in Appendix 2. Note that half of the 18-byte header has been blank filled. If a SAVALL file is to be stored from another background program, these fields must be changed to valid numbers for the storage process to complete successfully (see the source listing for STORALL or AFOS Programming Note No. 90).

C. Error Codes and Restrictions

The only restriction to either program is that files to be stored by STORALL may not be write protected. STORALL writes a new 18-byte header onto the disk file itself and initiates storage from this same file. Normally, the only files which will meet the validity requirements (Section II,B) are those created by SAVALL or the foreground SAVE command. None of these files will be write protected automatically. If a file is important enough to be protected, make a backup copy instead. AFOS archive files cannot be used as input to STORALL because they contain a filename extension (the hexadecimal version number).

Any operator input errors that can be detected are flagged, and the operator is given another chance. Unrecoverable system errors generate an error return to the CLI. In the latter case, SYSTEM ERROR will be typed by the program followed by the nature of the error typed by the CLI. To recover, you must correct the fault and start over.

LIST OF APPENDICES

- APPENDIX 1 - Sample interactive session with the operator
- APPENDIX 2 - Sample input and output for program SAVALL
- APPENDIX 3 - Program source listing: SAVALL
- APPENDIX 4 - Program source listing: STORALL
- APPENDIX 5 - Subroutine source listing: USTOR

APPENDIX 1

SAVALL
MOVE ALL PRODUCTS OF A GIVEN CATEGORY FROM THE DATABASE TO A DISK

ENTER DESTINATION DIRECTORY NAME (E.G., DP3) DP3
IF FILES ALREADY EXIST, DO YOU WANT TO OVERWRITE (Y/N)? N

AFTER A * PROMPT ENTER 1 CHARACTER FOR GRAPHICS,
3 CHARACTERS FOR A LOCAL NNN, 6 CHARACTERS FOR A REMOTE CCCNNN,
OR [RETURN] TO TERMINATE PROGRAM

*MIS
TOPMISCHK 01/04/82 21:18
TOPMISDDC 01/05/82 17:16
TOPMISGCK 04/14/82 15:59
NO CURRENT PRODUCT AVAILABLE: TOPMISGLD
TOPMISICT 04/12/82 20:18
TOPMISTOP 04/17/82 22:34

*
JOB COMPLETE
R
STORALL
MOVE AFDS PRODUCTS FROM A DISK TO THE DATABASE

ENTER SOURCE DIRECTORY NAME (E.G., DP3) DP3

AFTER A * PROMPT ENTER 1 CHARACTER FOR GRAPHICS,
3 CHARACTERS FOR A LOCAL NNN, 6 CHARACTERS FOR A REMOTE CCCNNN,
'ALL' FOR ALL VALID FILES, OR [RETURN] TO TERMINATE PROGRAM

*ALL
TOPMISCHK STORED
TOPMISDDC STORED
TOPMISGCK STORED
TOPMISTOP STORED
TOPMISICT STORED

*STLMIS
NO FILES MATCH SPECIFIER

*5
NO FILES MATCH SPECIFIER

*
JOB COMPLETE
R

APPENDIX 2

TOPSFPKS
FPUS1 KTOP 192210

STATE FORECAST FOR KANSAS
NATIONAL WEATHER SERVICE TOPEKA KS
510PM CDT SAT JUN 19 1982

CLEAR TONIGHT. LOWS IN THE 50S. PARTLY CLOUDY SUNDAY THROUGH MONDAY.
HIGHS BOTH SUNDAY AND MONDAY IN THE LOW TO MID 80S. LOWS SUNDAY NIGHT
FROM THE MID 50S NORTHWEST TO THE LOW 60S SOUTHEAST.

SCHURR

0	052117	050123	043120	045523	020040	020040	020040	020040	TOPSPFKS
10	020040	142600	043120	052523	030440	045524	047520	020061	E.FPUS1 KTOP 1
20	034462	031861	030015	005015	005123	052101	052105	020106	92210...STATE F
30	047522	042503	040523	052040	043117	051040	045501	047123	ORECAST FOR KANS
40	040523	006412	047101	052111	047516	040514	020127	042501	AS..NATIONAL WEA
50	052110	042522	020123	042522	053111	041505	020124	047520	THER SERVICE TOP
60	042513	040440	045523	006412	032461	030120	046440	041504	EKA KS..510PM CD
70	052040	051501	052040	045125	047040	030471	020061	034470	T SAT JUN 19 198
100	031015	005015	005103	046105	040522	020124	047516	044507	2....CLEAR TONIG
110	044124	027040	046117	053523	020111	047040	052110	042440	HT. LOWS IN THE
120	032460	051456	020120	040522	052114	054440	041514	047525	SOS. PARTLY CLOU
130	042131	020123	052516	042101	054440	052110	051117	052507	DY SUNDAY THROUG
140	044040	046517	047104	040531	027015	005110	044507	044123	H MONDAY...HIGHS
150	020102	047524	044040	051525	047104	040531	020101	047104	BOTH SUNDAY AND
160	020115	047516	042101	054440	044516	020124	044105	020114	MONDAY IN THE L
170	047527	020124	047440	046511	042040	034060	051456	020114	OW TO MID 00S. L
200	047527	051440	051525	047104	040531	020116	044507	044124	OWS SUNDAY NIGHT
210	006412	043122	047515	020124	044105	020115	044504	020065	..FROM THE MID 5
220	030123	020116	047522	052110	053505	051524	020124	047440	0S NORTHWEST TO
230	052110	042440	046117	053440	033060	051440	051517	052524	THE LOW 60S SOUT
240	044105	040523	052056	006412	006412	051503	044125	051122	HEAST....SCHURR
250	006412	101400	----	----	----	----	----	----	

APPENDIX 3

```

0001 SAVALL MACRO REV 06.00      04:17:54 04/18/82
                                .TITL   SAVALL      ;SAVE ALL (CCC)NNN OR NMCGRP× PRODUCTS
                                ;EVOLVED FROM PROGRAM 'BOB'
02                                .ENT    .MAIN
03                                .EXTN   .I,KSRCF,NXKF,RDBKF,NXBKF,KFILL,ADTG,DTG
04                                .EXTD   .FCAL
05
06                                ;
07                                ;   SAVE A COMPLETE CATEGORY OF AFOS PRODUCTS ONTO A SPECIFIED
08                                ;   DISK DIRECTORY
09                                ;
10                                ;   OUTPUT FILENAMES ARE OF THE FORM CCCNNNXXX. IF XXX IS ONE OR
11                                ;   TWO CHARACTERS, TRAILING SPACES ARE OMITTED.
12                                ;
13                                ;   PRODUCT NAMES AND CREATION TIMES ARE VERIFIED ON $TTO.
14                                ;
15                                ;   THE DESTINATION DIRECTORY IS INITIALIZED AT THE ONSET OF
16                                ;   THE PROGRAM.
17                                ;
18                                ;   RLDR SAVALL TOP.LB FORT.LB
19                                ;
20                                ;   PROGRAMMER:  WARREN E. SUNKEL, WSFO TOPEKA
21                                ;
22      000001      .TXTM   1
23                                .ZREL
24 00000-000575' .SPCH: SPCHR
25 00001-000000 COUNT: 0
26 00002-000532' .ERR:  ERR
27 00003-001440" .DP:  DP*2
28 00004-003134" .CR:  CR*2
29 00005-000000 FLAG:  0      ;--1 IF A PRODUCT IS OUTPUT
30
31                                .NREL
32 00000'002172" .TTF:  TTF*2
33 00001'000000 TTC:  0      ;CHANNEL FOR $TTI
34 00002'002216" .M1:  M1*2
35 00003'002322" .M2:  M2*2
36 00004'002406" .M3:  M3*2
37 00005'002504" .M4:  M4*2
38 00006'002566" .M5:  M5*2
39 00007'002670" .M6:  M6*2
40 00010'000131 C131:  131
41 00011'000045 C45:  45
42
43      B0:
44 00012'020777   LDA    0,C45      ;DIRECTORY ALREADY INITIALIZED
45 00013'142404   SUB    2,0,SZR
46 00014'002002-   JMP    0,ERR      ;SOME OTHER ERROR
47 00015'001400   JMP    0,3
48
49 00016'001510" .BUF:  BUF*2
50 00017'000000   0      ;STACK LENGTH WORD
51      .MAIN:
52 00020'006017   .SYSTEM      ;OPEN A CHANNEL TO $TTI
53 00021'021052   .GCHN
54 00022'002002-   JMP    0,ERR
55 00023'020755   LDA    0,.TTF      ;$TTI FILENAME
56 00024'126400   SUB    1,1
57 00025'006017   .SYSTEM
58 00026'014077   .OPEN    77
59 00027'002002-   JMP    0,ERR
60 00030'050751   STA    2,TTC      ;SAVE CHANNEL
61
62 00031'030751   LDA    2,.M1      ;PROGRAM TITLE

```

```

0002 SAVAL
01 00032'006000-   JSR    @.SPCH
02 00033'030004-   LDA    2,.CR           ;LF
03 00034'006000-   JSR    @.SPCH
04 00035'030746   LDA    2,.M2           ;DESTINATION DIRECTORY
05 00036'006000-   JSR    @.SPCH
06 00037'020003-   LDA    0,.DP
07 00040'030741   LDA    2,TTC
08 00041'006017   .SYSTEM
09 00042'015477   .RDL    77
10 00043'002002-   JMP    @.ERR
11 00044'104110   SBI    1,1           ;STORE NULL OVER CR
12 00045'107000   ADD    0,1
13 00046'152400   SUB    2,2
14 00047'133010   STB    1,2
15 00050'126400   SUB    1,1           ;INITIALIZE DIRECTORY
16 00051'006017   .SYSTEM
17 00052'004000   .INIT
18 00053'004737   JSR    B0
19 00054'030730   LDA    2,.M3           ;OVERWRITE?
20 00055'006000-   JSR    @.SPCH
21 00056'020740   LDA    0,.BUF
22 00057'030722   LDA    2,TTC
23 00060'006017   .SYSTEM
24 00061'015477   .RDL    77
25 00062'002002-   JMP    @.ERR
26 00063'030733   LDA    2,.BUF
27 00064'142710   LDB    2,0
28 00065'024723   LDA    1,C131           ;Y
29 00066'152400   SUB    2,2
30 00067'122405   SUB    1,0.SNR
31 00070'152000   ADC    2,2           ;=-1 TO REPLACE
32 00071'050574   STA    2,RPL
33 00072'030004-   LDA    2,.CR           ;LF
34 00073'006000-   JSR    @.SPCH
35 00074'030711   LDA    2,.M4           ;AFTER *...
36 00075'006000-   JSR    @.SPCH
37 00076'030710   LDA    2,.M5           ;3 CHARS...
38 00077'006000-   JSR    @.SPCH
39 00100'030707   LDA    2,.M6           ;OR [RETURN]...
40 00101'006000-   JSR    @.SPCH
41 00102'000410   JMP    B1.1
42
43 00103'020005-   LDA    0,FLAG           ;CHECK FOR ANY OUTPUT
44 00104'101004   MOV    0,0.SZR
45 00105'000405   JMP    B1.1           ;YES, SKIP MESSAGE
46
47 00106'030004-   LDA    2,.CR
48 00107'006000-   JSR    @.SPCH
49 00110'030565   LDA    2,.M7           ;NO PRODUCTS...
50 00111'006000-   JSR    @.SPCH
51
52 00112'030004-   LDA    2,.CR           ;TYPE LINE FEED AND PROMPT
53 00113'006000-   JSR    @.SPCH
54 00114'030563   LDA    2,.PMT
55 00115'006000-   JSR    @.SPCH
56
57 00116'020700   LDA    0,.BUF           ;READ OPERATOR INPUT
58 00117'030662   LDA    2,TTC
59 00120'006017   .SYSTEM
60 00121'015477   .RDL    77
61 00122'002002-   JMP    @.ERR
62 00123'104110   SBI    1,1           ;SUBTRACT CR CHARACTER

```

```

0003 SAVAL
01 00124'125005      MOV      1,1,SNR
02 00125'002537      JMP      @.B12      ;CLOSE ON NO INPUT
03 00126'102000      ADC      0,0
04 00127'040571      STA      0,GFLAG      ;SET GRAPHIC FLAG TO TRUE
05 00130'102520      SUBZL     0,0
06 00131'122405      SUB      1,0,SNR      ;CHECK 1-CHARACTER INPUT
07 00132'000414      JMP      B2
08 00133'102400      SUB      0,0
09 00134'040564      STA      0,GFLAG      ;SET GRAPHIC FLAG TO FALSE
10 00135'020531      LDA      0,C3      ;CHECK 3-CHARACTER INPUT
11 00136'122405      SUB      1,0,SNR
12 00137'000407      JMP      B2
13 00140'020527      LDA      0,C6      ;CHECK 6-CHARACTER INPUT
14 00141'122405      SUB      1,0,SNR
15 00142'000404      JMP      B2
16 00143'030533      LDA      2,.M9      ;INPUT ERROR
17 00144'006000-      JSR      @.SPCH
18 00145'000736      JMP      B1
19
B2:
20 00146'034650      LDA      3,.BUF      ;ADD XXX FOR NULL KEY
21 00147'137000      ADD      1,3
22 00150'020530      LDA      0,C4
23 00151'040001-      STA      0,COUNT
24 00152'030516      LDA      2,.Z      ;000
25 00153'020545      LDA      0,GFLAG
26 00154'101004      MOV      0,0,SZR
27 00155'151400      INC      2,2      ;ONLY TWO NULLS FOR GRAPHICS
28
B3:
29 00156'142710      LDB      2,0
30 00157'163010      STB      3,0
31 00160'151400      INC      2,2
32 00161'175400      INC      3,3
33 00162'014001-      DSZ      COUNT
34 00163'000773      JMP      B3
35
36 00164'006001$      JSR      @.FCAL      ;FILL KEY TO 9 CHARACTERS
37 00165'077777      KFILL
38 00166'000002      2
39 00167'000644'      BUF
40 00170'000273'      IER
41
42 00171'006001$      JSR      @.FCAL      ;FIND NULL KEY IN DATAKEY0
43 00172'077777      KSRCF
44 00173'000003      3
45 00174'000644'      BUF
46 00175'000656'      KREC
47 00176'000273'      IER
48
49 00177'102400      SUB      0,0      ;SET OUTPUT FLAG TO FALSE
50 00200'040005-      STA      0,FLAG
51
B4:
52 00201'006001$      JSR      @.FCAL      ;SEARCH NEXT KEY IN DATAKEY0
53 00202'077777      NXKF
54 00203'000002      2
55 00204'000656'      KREC
56 00205'000273'      IER
57
58 00206'014465      DSZ      IER
59 00207'000674      JMP      B1      ;NO NEXT KEY
60
61 00210'020457      LDA      0,C6      ;COMPARE NEW KEY TO CURRENT CATEGORY
62 00211'024507      LDA      1,GFLAG

```

```

0004 SAVAL
01 00212'125004      MOV      1,1,SZR
02 00213'101400      INC       0,0           ;7 CHARACTERS FOR GRAPHICS
03 00214'040001-     STA       0,COUNT
04 00215'030601      LDA       2,.BUF
05 00216'034543      LDA       3,.KREC
06
07 00217'142710      LDB       2,0
08 00220'166710      LDB       3,1
09 00221'122404      SUB       1,0,SZR
10 00222'000661      JMP       B1           ;HAVE REACHED NEW CATEGORY
11 00223'151400      INC       2,2
12 00224'175400      INC       3,3
13 00225'014001-     DSZ       COUNT
14 00226'000771      JMP       B5
15
16 00227'102000      ADC       0,0           ;SET PRODUCT FLAG TO TRUE
17 00230'040005-     STA       0,FLAG
18 00231'020525      LDA       0,D9           ;STORE NULL AT END OF KREC KEY
19 00232'034527      LDA       3,.KREC
20 00233'117000      ADD       0,3
21 00234'102400      SUB       0,0
22 00235'163010      STB       3,0
23
24 00236'006001$     JSR       @.FCAL           ;READ BLOCK 0
25 00237'077777      RDBKF
26 00240'000003      3
27 00241'000301'     C0
28 00242'000675'     BLOCK
29 00243'000273'     IER
30
31 00244'014427      DSZ       IER
32 00245'000410      JMP       B5.1           ;NO PRODUCT AVAILABLE
33
34 00246'006001$     JSR       @.FCAL           ;SAVE DTG OF PRODUCT
35 00247'077777      ADTG
36 00250'000003      3
37 00251'000675'     BLOCK
38 00252'000302'     DATE
39 00253'000305'     DATE+3
40 00254'000446      JMP       B6
41
42 00255'030417      LDA       2,.M12           ;NO PRODUCT AVAILABLE
43 00256'006000-     JSR       @.SPCH
44 00257'030502      LDA       2,.KREC
45 00260'006000-     JSR       @.SPCH
46 00261'030004-     LDA       2,.CR
47 00262'006000-     JSR       @.SPCH
48 00263'000716      JMP       B4           ;FETCH NEXT KEY
49
50 00264'000541'.B12: B12
51 00265'000000 RPL:  0           ;=-1 TO REPLACE EXISTING DISK FILES
52 00266'000003 C3:   3
53 00267'000006 C6:   6
54 00270'000562".Z:  .+1*2
55 00271'030060      .TXT      "000"
56      030000
57 00273'000000 IER:  0           ;ERROR RETURN FROM SUBROUTINE CALLS
58 00274'003072".M12: M12*2
59 00275'002732".M7:  M7*2
60 00276'003020".M9:  M9*2
61 00277'003132".PMT: PMT*2
62 00300'000004 C4:   4

```

```

0005 SAVAL
01 00301'000000 C0:      0
02 00302'000005 DATE:    .BLK      5      ;DATE/TIME GROUP
03 00307'000017 D15:      15.
04 00310'000072 C72:      72
05 00311'000040 C40:      40
06 00312'000000 CH:       0      ;OUTPUT CHANNEL
07 00313'002204".SP:      SPC*2
08 00314'000012 D10:      10.
09 00315'000027 D23:      23.
10 00316'000400 D256:     256.
11 00317'000011 C11:      11
12 00320'000000 GFLAG:    0      ;=-1 FOR GRAPHICS
13 00321'001464".FN:      FN*2
14
15      B6:
16 00322'020765          LDA      0,D15          ;CREATE FILENAME
17 00323'040001-          STA      0,COUNT
18 00324'030003-          LDA      2,.DP
19 00325'034774          LDA      3,.FN
20      B6.1:
21 00326'142710          LDB      2,0            ;FIND END OF DIRECTORY NAME
22 00327'101005          MOV      0,0,SNR
23 00330'000406          JMP      B6.2
24 00331'163010          STB      3,0
25 00332'151400          INC      2,2
26 00333'175400          INC      3,3
27 00334'014001-          DSZ      COUNT
28 00335'000771          JMP      B6.1
29      B6.2:
30 00336'020752          LDA      0,C72          ; :
31 00337'163010          STB      3,0
32 00340'175400          INC      3,3
33 00341'020753          LDA      0,D10          ;MOVE KEY AND NULL FROM KREC
34 00342'040001-          STA      0,COUNT
35 00343'030416          LDA      2,.KREC
36 00344'024745          LDA      1,C40
37      B6.3:
38 00345'142710          LDB      2,0
39 00346'122415          SUB#     1,0,SNR
40 00347'102400          SUB      0,0            ;STORE NULLS OVER BLANKS
41 00350'163010          STB      3,0
42 00351'151400          INC      2,2
43 00352'175400          INC      3,3
44 00353'014001-          DSZ      COUNT
45 00354'000771          JMP      B6.3
46 00355'000406          JMP      B6.4
47
48 00356'000011 D9:       9.
49 00357'000016 D14:      14.
50 00360'000015 C15:      15
51 00361'001534".KREC:    KREC*2
52 00362'001572".BLOC:    BLOCK*2
53
54      B6.4:
55 00363'020736          LDA      0,.FN          ;INITIALIZE OUTPUT
56 00364'024701          LDA      1,RPL          ;CHECK REPLACE FLAG
57 00365'125005          MOV      1,1,SNR
58 00366'000404          JMP      B7.            ;DO NOT OVERWRITE
59 00367'006017          .SYSTEM
60 00370'000400          .DELET
61 00371'000401          JMP      .+1
62      B7:

```

```

0006 SAVAL
01 00372'006017 .SYSTEM
02 00373'007000 .CRAND
03 00374'000456 JMP B9
04 00375'006017 .SYSTEM
05 00376'021052 .GCHN
06 00377'002002- JMP @.ERR
07 00400'126400 SUB 1,1
08 00401'006017 .SYSTEM
09 00402'014077 .OPEN 77
10 00403'002002- JMP @.ERR
11 00404'050706 STA 2,CH ;OUTPUT CHANNEL
12
13 00405'020754 LDA @,.KREC ;OUTPUT HEADER
14 00406'024750 LDA 1,D9 ;9 BYTES OF KEY
15 00407'006017 .SYSTEM
16 00410'016477 .WRS 77
17 00411'002002- JMP @.ERR
18 00412'024744 LDA 1,D9 ;AND 9 BLANKS
19 00413'020700 LDA @,.SP
20 00414'006017 .SYSTEM
21 00415'016477 .WRS 77
22 00416'002002- JMP @.ERR
23
24 00417'020743 LDA @,.BLOC ;OUTPUT BLOCK 0
25 00420'034675 LDA 3,D23 ;23 BYTES OF HEADER
26 00421'163000 ADD 3,0
27 00422'024674 LDA 1,D256
28 00423'166400 SUB 3,1 ;REMAINING BYTES
29 00424'004522 JSR SCAN ;SCAN FOR ETX
30 00425'006017 .SYSTEM
31 00426'016477 .WRS 77
32 00427'002002- JMP @.ERR
33
34 00430'006001$ B8: JSR @.FCAL ;READ NEXT BLOCK
35 00431'077777 NXBKF
36 00432'000002 2
37 00433'000675' BLOCK
38 00434'000273' IER
39
40 00435'014636 DSZ IER
41 00436'000426 JMP B10 ;END OF PRODUCT
42
43 00437'020723 LDA @,.BLOC
44 00440'034640 LDA 3,C4 ;BLOCK N HEADER BYTES
45 00441'163000 ADD 3,0
46 00442'024654 LDA 1,D256
47 00443'166400 SUB 3,1 ;REMAINING BYTES
48 00444'004502 JSR SCAN ;SCAN FOR ETX
49 00445'030645 LDA 2,CH
50 00446'006017 .SYSTEM
51 00447'016477 .WRS 77
52 00450'002002- JMP @.ERR
53 00451'000757 JMP B8 ;FETCH NEXT BLOCK
54
55 00452'020645 B9: LDA @,C11 ;FILE ALREADY EXISTS
56 00453'142404 SUB 2,0,SZR
57 00454'002002- JMP @.ERR ;SOME OTHER ERROR
58 00455'030515 LDA 2,.MB
59 00456'006000- JSR @.SPCH
60 00457'030642 LDA 2,.FN ;BP TO FILENAME
61 00460'006000- JSR @.SPCH
62 00461'030004- LDA 2,.CR

```

```

0007 SAVAL
01 00462'006000- JSR @.SPCH
02 00463'002506 JMP @.B4 ;FETCH NEXT KEY
03 B10:
04 00464'030626 LDA 2,CH ;CLOSE OUTPUT FILE
05 00465'006017 .SYSTEM
06 00466'014477 .CLOSE 77
07 00467'002002- JMP @.ERR
08
09 00470'020610 LDA 0,C4 ;TYPE PRODUCT KEY AND DTG
10 00471'040001- STA 0,COUNT
11 00472'030667 LDA 2,.KREC
12 00473'020663 LDA 0,D9
13 00474'113000 ADD 0,2
14 00475'020614 LDA 0,C40 ;STORE SPACES AFTER KEY
15 00476'143010 STB 2,0
16 00477'151400 INC 2,2
17 00500'014001- DSZ COUNT
18 00501'000775 JMP .-3
19
20 00502'102520 SUBZL 0,0
21 00503'040465 STA 0,N
22 00504'006001$ JSR @.FCAL ;ENCODE DTG
23 00505'077777 DTG
24 00506'000004 4
25 00507'000302' DATE
26 00510'000305' DATE+3
27 00511'000675' BLOCK
28 00512'000570' N
29
30 00513'020644 LDA 0,D14 ;PACK AND MOVE
31 00514'040001- STA 0,COUNT
32 00515'034645 LDA 3,.BLOC
33 00516'175220 MOVZR 3,3
34 B11:
35 00517'021400 LDA 0,0,3
36 00520'143010 STB 2,0
37 00521'151400 INC 2,2
38 00522'175400 INC 3,3
39 00523'014001- DSZ COUNT
40 00524'000773 JMP B11
41 00525'020633 LDA 0,C15
42 00526'143010 STB 2,0
43
44 00527'030632 LDA 2,.KREC ;TYPE PRODUCT ID
45 00530'006000- JSR @.SPCH
46 00531'002440 JMP @.B4 ;SEARCH FOR NEXT KEY
47 ERR:
48 00532'050001- STA 2,COUNT
49 00533'030440 LDA 2,.M10 ;SYSTEM ERROR
50 00534'006000- JSR @.SPCH
51 00535'030001- LDA 2,COUNT
52 00536'006017 .SYSTEM
53 00537'006400 .ERTN
54 00540'000776 JMP .-2
55 B12:
56 00541'030433 LDA 2,.M11 ;JOB COMPLETE
57 00542'006000- JSR @.SPCH
58 00543'006017 .SYSTEM
59 00544'004400 .RTN
60 00545'000776 JMP .-2
61 SCAN:
62 00546'054420 STA 3,S3 ;SCAN OUTPUT BUFFER FOR ETX (203)

```

```

0008 SAVAL
01 00547'040416 STA 0,S0
02 00550'115000 MOV 0,3
03 00551'044001- STA 1,COUNT
04 00552'024415 LDA 1,ETX
05 00553'162710 S.1: LDB 3,0
06 00554'175400 INC 3,3
07 00555'122405 SUB 1,0,SNR
08 00556'000403 JMP S.2
09 00557'014001- DSZ COUNT
10 00560'000773 JMP S.1
11 00561'165000 S.2: MOV 3,1 ;CALCULATE NUMBER OF BYTES TO WRITE
12 00562'020403 LDA 0,S0
13 00563'106400 SUB 0,1
14 00564'002402 JMP 0S3
15 00565'000000 S0: 0
16 00566'000000 S3: 0
17 00567'000203 ETX: 203
18
19 00570'000000 N: 0 ;COUNTER FOR DTG SUBROUTINE
20 00571'000201'.B4: B4
21 00572'002770".M8: M8*2
22 00573'003036".M10: M10*2
23 00574'003054".M11: M11*2
24 SPCHR:
25 00575'054422 STA 3,SAV3 ;$TTO OUTPUT ROUTINE
26 00576'142710 LDB 2,0 ;CHARACTER TO AC0
27 00577'101005 MOV 0,0,SNR ;CHECK FOR NULL
28 00600'002417 JMP 0SAV3 ;YES,RETURN
29 00601'006017 .SYSTM
30 00602'010000 .PCHAR
31 00603'000401 JMP .+1
32 00604'151400 INC 2,2
33 00605'024410 LDA 1,S15 ;CHECK FOR CR
34 00606'106404 SUB 0,1,SZR
35 00607'000767 JMP SPCHR+1 ;NO, CONTINUE
36 00610'020406 LDA 0,S12 ;LINE FEED
37 00611'006017 .SYSTM
38 00612'010000 .PCHAR
39 00613'000401 JMP .+1
40 00614'002403 JMP 0SAV3 ;AND RETURN
41 00615'000015 S15: 15
42 00616'000012 S12: 12
43 00617'000000 SAV3: 0
44
45 00620'000012 DP: .BLK 10. ;DIRECTORY SPECIFIER
46 00632'000012 FN: .BLK 10. ;OUTPUT FILENAME
47 00644'000012 BUF: .BLK 10. ;OPERATOR INPUT/NULL KEY
48 00656'000017 KREC: .BLK 17
49 00675'000200 BLOCK: .BLK 120.
50 01075'042120 TTF: .TXT "DP0:$TT1"
51 01102'020040 SPC: .TXT " "
52 01107'046517 M1: .TXT "MOVE ALL PRODUCTS OF A GIVEN CATEGORY FROM THE DATABASE
53 01142'042440 TO A DISK<15>"
54 01151'042516 M2: .TXT "ENTER DESTINATION DIRECTORY NAME (E.G., DP3) "
55 01203'044506 M3: .TXT "IF FILES ALREADY EXIST, DO YOU WANT TO OVERWRITE (Y/N)? "
56 01242'040506 M4: .TXT "AFTER A * PROMPT ENTER 1 CHARACTER FOR GRAPHICS,<15>"
57 01273'031440 M5: .TXT "3 CHARACTERS FOR A LOCAL NNN, 6 CHARACTERS FOR A
58 01323'020122 REMOTE CCCNN,<15>"
59 01334'047522 M6: .TXT "OR [RETURN] TO TERMINATE PROGRAM<15>"
60 01355'047117 M7: .TXT "NO PRODUCTS MATCH SPECIFIER<15>"
61 01374'043111 M8: .TXT "FILE ALREADY EXISTS: "
62 01410'044516 M9: .TXT "INPUT ERROR<15>"

```

```
0009 SAVAL
01 01417'051531 M10: .TXT "SYSTEM ERROR<15>"
02 01426'045117 M11: .TXT "JOB COMPLETE<15>"
03 01435'047117 M12: .TXT "NO CURRENT PRODUCT AVAILABLE: "
04 01455'025000 PMT: .TXT "*"
05 01456'006400 CR: .TXT "<15>"
06 000000 .NOLOC 0
07 .END
```

*k00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0010 SAVAL

ADTG	000247'	XN	1/04	4/35					
B0	000012'		1/43	2/18					
B1	000103'		2/42	3/18	3/59	4/10			
B10	000464'		6/41	7/03					
B11	000517'		7/34	7/40					
B12	000541'		4/50	7/55					
B1.1	000112'		2/41	2/45	2/51				
B2	000146'		3/07	3/12	3/15	3/19			
B3	000156'		3/28	3/34					
B4	000201'		3/51	4/48	8/20				
B5	000217'		4/06	4/14					
B5.1	000255'		4/32	4/41					
B6	000322'		4/40	5/15					
B6.1	000326'		5/20	5/28					
B6.2	000336'		5/23	5/29					
B6.3	000345'		5/37	5/45					
B6.4	000363'		5/46	5/54					
B7	000372'		5/58	5/62					
B8	000430'		6/33	6/53					
B9	000452'		6/03	6/54					
BLOCK	000675'		4/28	4/37	5/52	6/37	7/27	8/49	
BUF	000644'		1/49	3/39	3/45	8/47			
C0	000301'		4/27	5/01					
C11	000317'		5/11	6/55					
C131	000010'		1/40	2/28					
C15	000360'		5/50	7/41					
C3	000266'		3/10	4/52					
C4	000300'		3/22	4/62	6/44	7/09			
C40	000311'		5/05	5/36	7/14				
C45	000011'		1/41	1/44					
C6	000267'		3/13	3/61	4/53				
C72	000310'		5/04	5/30					
CH	000312'		5/06	6/11	6/49	7/04			
COUNT	000001-		1/25	3/23	3/33	4/03	4/13	5/17	5/27
			5/34	5/44	7/10	7/17	7/31	7/39	7/48
			7/51	8/03	8/09				
CR	001456'		1/28	9/05					
D10	000314'		5/08	5/33					
D14	000357'		5/49	7/30					
D15	000307'		5/03	5/16					
D23	000315'		5/09	6/25					
D256	000316'		5/10	6/27	6/46				
D9	000356'		4/18	5/48	6/14	6/18	7/12		
DATE	000302'		4/38	4/39	5/02	7/25	7/26		
DP	000620'		1/27	8/45					
DTG	000505'	XN	1/04	7/23					
ERR	000532'		1/26	7/47					
ETX	000567'		8/04	8/17					
FLAG	000005-		1/29	2/43	3/50	4/17			
FN	000632'		5/13	8/46					
GFLAG	000320'		3/04	3/09	3/25	3/62	5/12		
IER	000273'		3/40	3/47	3/56	3/58	4/29	4/31	4/57
			6/38	6/40					
KFILL	000165'	XN	1/04	3/37					
KREC	000656'		3/46	3/55	5/51	8/48			
KSRCF	000172'	XN	1/04	3/43					
M1	001107'		1/34	8/52					
M10	001417'		8/22	9/01					
M11	001426'		8/23	9/02					
M12	001435'		4/58	9/03					
M2	001151'		1/35	8/54					
M3	001203'		1/36	8/55					

0011 SAVAL

M4	001242'	1/37	8/56						
M5	001273'	1/38	8/57						
M6	001334'	1/39	8/59						
M7	001355'	4/59	8/60						
M8	001374'	8/21	8/61						
M9	001410'	4/60	8/62						
N	000570'	7/21	7/28	8/19					
NXBKF	000431' XN	1/04	6/35						
NXKF	000202' XN	1/04	3/53						
PMT	001455'	4/61	9/04						
RDBKF	000237' XN	1/04	4/25						
RPL	000265'	2/32	4/51	5/56					
S0	000565'	8/01	8/12	8/15					
S12	000616'	8/36	8/42						
S15	000615'	8/33	8/41						
S3	000566'	7/62	8/14	8/16					
SAV3	000617'	8/25	8/28	8/40	8/43				
SCAN	000546'	6/29	6/48	7/61					
SPC	001102'	5/07	8/51						
SPCHR	000575'	1/24	8/24	8/35					
S.1	000553'	8/05	8/10						
S.2	000561'	8/08	8/11						
TTC	000001'	1/33	1/60	2/07	2/22	2/58			
TTF	001075'	1/32	8/50						
.B12	000264'	3/02	4/50						
.B4	000571'	7/02	7/46	8/20					
.BLOC	000362'	5/52	6/24	6/43	7/32				
.BUF	000016'	1/49	2/21	2/26	2/57	3/20	4/04		
.CR	000004-	1/28	2/02	2/33	2/47	2/52	4/46	6/62	
.DP	000003-	1/27	2/06	5/18					
.ERR	000002-	1/26	1/46	1/54	1/59	2/10	2/25	2/61	
		6/06	6/10	6/17	6/22	6/32	6/52	6/57	
		7/07							
.FCAL	000001\$ XD	1/05	3/36	3/42	3/52	4/24	4/34	6/34	
		7/22							
.FN	000321'	5/13	5/19	5/55	6/60				
.I	077777' XN	1/04							
.KREC	000361'	4/05	4/19	4/44	5/35	5/51	6/13	7/11	
		7/44							
.M1	000002'	1/34	1/62						
.M10	000573'	7/49	8/22						
.M11	000574'	7/56	8/23						
.M12	000274'	4/42	4/58						
.M2	000003'	1/35	2/04						
.M3	000004'	1/36	2/19						
.M4	000005'	1/37	2/35						
.M5	000006'	1/38	2/37						
.M6	000007'	1/39	2/39						
.M7	000275'	2/49	4/59						
.M8	000572'	6/58	8/21						
.M9	000276'	3/16	4/60						
.MAIN	000020' EN	1/03	1/51						
.PMT	000277'	2/54	4/61						
.SP	000313'	5/07	6/19						
.SPCH	000000-	1/24	2/01	2/03	2/05	2/20	2/34	2/36	
		2/38	2/40	2/48	2/50	2/53	2/55	3/17	
		4/43	4/45	4/47	6/59	6/61	7/01	7/45	
		7/50	7/57						
.TTF	000000'	1/32	1/55						
Z	000270'	3/24	4/54						

APPENDIX 4

```

0001 STORA MACRO REV 06.00          04:14:25 04/18/82
      .TITL   STORALL           ;STORE ALL DISK FILES INTO THE DATABASE
02      .ENT   .MAIN
03      .EXTD  .FCAL
04      .EXTN  .I,USTOR,KFILL
05      ;
06      ;   STORE FILES FROM A SPECIFIED DIRECTORY INTO THE AFOS DATABASE
07      ;
08      ;   FOR A FILE TO BE STORED, THE FOLLOWING CONDITIONS MUST BE MET:
09      ;
10      ;       THE FILENAME MUST BE 7 TO 9 CHARACTERS WITH NO
11      ;       EXTENSION.
12      ;
13      ;       THE FIRST 7 TO 9 CHARACTERS OF THE FILE (AFOS HEADER)
14      ;       MUST MATCH THE FILENAME.
15      ;
16      ;       THE APPROPRIATE PART OF THE FILENAME MUST MATCH THE
17      ;       OPERATOR'S TEMPLATE.
18      ;
19      ;   THE ADDRESSEE, DTG, PRIORITY AND TYPE ARE UPDATED BY THIS PROGRAM
20      ;   BEFORE THE FILE IS STORED.  PROGRAM LOOPS UNTIL STORAGE IS COMPLETE.
21      ;
22      ;   THE SOURCE DIRECTORY IS INITIALIZED AT THE ONSET OF THE PROGRAM.
23      ;
24      ;   RLDR STORALL USTOR TOP.LB FORT.LB
25      ;
26      ;   PROGRAMMER:  WARREN E. SUNKEL, WSFO TOPEKA
27      ;
28      000001      .TXTM   1
29      .ZREL
30 00000-000000 COUNT: 0
31 00001-000556'.ERR: ERR
32 00002-000577'.SPCH: SPCHR
33 00003-000566'.MCHR: MCHR
34 00004-000000 ALL: 0           ;=-1 TO STORE ALL FILES
35 00005-000000 GFLAG: 0        ;=-1 FOR GRAPHICS
36 00006-000000 FLAG: 0        ;=-1 IF A PRODUCT IS STORED
37 00007-000006 C6: 6
38      .NREL
39 00000'000000      0           ;STACK LENGTH WORD
40      .MAIN:
41 00001'006017      .SYSTM      ;OPEN A CHANNEL TO $TTI
42 00002'021052      .GCHN
43 00003'002001-      JMP        @.ERR
44 00004'020536      LDA        0,.TTF
45 00005'126400      SUB        1,1
46 00006'006017      .SYSTM
47 00007'014077      .OPEN      77
48 00010'002001-      JMP        @.ERR
49 00011'050511      STA        2,TTC
50
51 00012'030516      LDA        2,.M1           ;PROGRAM TITLE
52 00013'006002-      JSR        @.SPCH
53 00014'030511      LDA        2,.CR
54 00015'006002-      JSR        @.SPCH
55 00016'030513      LDA        2,.M2           ;SOURCE DIRECTORY?
56 00017'006002-      JSR        @.SPCH
57 00020'020507      LDA        0,.DP           ;READ INPUT
58 00021'030501      LDA        2,TTC
59 00022'006017      .SYSTM
60 00023'015477      .RDL        77
61 00024'002001-      JMP        @.ERR
62 00025'104110      SBI        1,1           ;STORE NULL AT END OF DIRECTORY NAME

```

```

0002 STORA
01 00026'107000      ADD      0,1
02 00027'152400      SUB      2,2
03 00030'133010      STB      1,2
04 00031'126400      SUB      1,1
05 00032'006017      .SYSTEM      ; INITIALIZE DIRECTORY
06 00033'004000      .INIT
07 00034'004456      JSR      B1.2
08 00035'030470      LDA      2,.CR      ;LF
09 00036'006002-      JSR      0,SPCH
10 00037'030473      LDA      2,.M4      ;AFTER *...
11 00040'006002-      JSR      0,SPCH
12 00041'030472      LDA      2,.M5      ;3 CHARACTERS...
13 00042'006002-      JSR      0,SPCH
14 00043'030471      LDA      2,.M6      ;OR ALL...
15 00044'006002-      JSR      0,SPCH
16 00045'000410      JMP      B1.1
17
18 00046'020006-      LDA      0,FLAG      ;CHECK FOR ANY PRODUCTS
19 00047'101004      MOV      0,0,SZR
20 00050'000405      JMP      B1.1
21 00051'030454      LDA      2,.CR
22 00052'006002-      JSR      0,SPCH
23 00053'030462      LDA      2,.M7      ;NO FILES...
24 00054'006002-      JSR      0,SPCH
25
26 00055'102400      SUB      0,0
27 00056'040006-      STA      0,FLAG      ;PRODUCT FLAG TO FALSE
28 00057'030446      LDA      2,.CR      ;TYPE LINE FEED AND PROMPT
29 00060'006002-      JSR      0,SPCH
30 00061'030456      LDA      2,.PMT
31 00062'006002-      JSR      0,SPCH
32 00063'020524      LDA      0,.TP
33 00064'030436      LDA      2,TTC      ;READ FILENAME TEMPLATE
34 00065'006017      .SYSTEM
35 00066'015477      .RDL      77
36 00067'002001-      JMP      0,ERR
37 00070'104110      SBI      1,1
38 00071'125005      MOV      1,1,SNR      ;SUBTRACT CR CHARACTER
39 00072'002432      JMP      0,B18      ;CLOSE ON NO INPUT
40 00073'102400      SUB      0,0
41 00074'040005-      STA      0,GFLAG      ;SET GRAPHIC FLAG TO FALSE
42 00075'040004-      STA      0,ALL      ;SET ALL TO FALSE
43 00076'102520      SUBZL      0,0
44 00077'122405      SUB      1,0,SNR      ;CHECK 1-CHARACTER INPUT
45 00100'000444      JMP      B2      ;BRANCH TO GRAPHICS ROUTINE
46 00101'020417      LDA      0,C3      ;CHECK 3-CHARACTER INPUT
47 00102'122405      SUB      1,0,SNR
48 00103'000454      JMP      B3      ;BRANCH TO 3-CHARACTER ROUTINE
49 00104'020007-      LDA      0,C6      ;CHECK 6-CHARACTER INPUT
50 00105'122405      SUB      1,0,SNR
51 00106'000502      JMP      B4      ;BRANCH TO 6-CHARACTER ROUTINE
52 00107'030427      LDA      2,.M9      ;INPUT ERROR
53 00110'006002-      JSR      0,SPCH
54 00111'000744      JMP      B1.1
55
56 00112'020404      LDA      0,C45      ;DIRECTORY ALREADY INITIALIZED
57 00113'142404      SUB      2,0,SZR
58 00114'002001-      JMP      0,ERR      ;SOME OTHER ERROR
59 00115'001400      JMP      0,3
60
61 00116'000045 C45: 45
62 00117'000015 C15: 15

```

```

0003 STORA
01 00120'000003 C3:      3
02 00121'000060 C60:    60
03 00122'000000 TTC:      0      ;$TTI CHANNEL
04 00123'002624".A:      A*2      ;"ALL"
05 00124'000550".B18:    B18
06 00125'000254".CR:      .+1*2
07 00126'006400          .TXT      "<15>"
08 00127'002470".DP:    DP*2      ;SOURCE DIRECTORY
09 00130'002630".M1:    M1*2
10 00131'002710".M2:    M2*2
11 00132'002766".M4:    M4*2
12 00133'003050".M5:    M5*2
13 00134'003152".M6:    M6*2
14 00135'003250".M7:    M7*2
15 00136'003302".M9:    M9*2
16 00137'000300".PMT:    .+1*2
17 00140'025000          .TXT      "*"
18 00141'003354".SD:      SD*2
19 00142'003376".TTF:    TTF*2
20 00143'003404".NMC:    NMC*2      ;"NMC GPH"
21
22          B2:
23 00144'030007-          LDA      2,C6      ;MOVE 'NMC GPH' TO TEMPLATE
24 00145'034442          LDA      3,.TP
25 00146'162710          LDB      3,0      ;SHIFT FIRST CHARACTER
26 00147'157000          ADD      2,3
27 00150'163010          STB      3,0
28 00151'020772          LDA      0,.NMC
29 00152'024435          LDA      1,.TP
30 00153'006003-          JSR      0,MCHR
31 00154'102000          ADC      0,0
32 00155'040005-          STA      0,GFLAG      ;SET GRAPHIC FLAG TO TRUE
33 00156'000432          JMP      B4
34          B3:
35 00157'020430          LDA      0,.TP      ;CHECK FOR ALL FILES
36 00160'024743          LDA      1,.A
37 00161'030737          LDA      2,C3
38 00162'004550          JSR      MATCH
39 00163'040004-          STA      0,ALL      ;SET 'ALL' FLAG
40 00164'101004          MOV      0,0,SZR
41 00165'000423          JMP      B4      ;JUMP IF TRUE
42 00166'030421          LDA      2,.TP      ;OTHERWISE INSERT THE LOCAL CCC
43 00167'020731          LDA      0,C3
44 00170'113000          ADD      0,2
45 00171'020730          LDA      0,C60      ;STORE A CHARACTER
46 00172'143010          STB      2,0
47 00173'151400          INC      2,2
48 00174'102400          SUB      0,0      ;AND A NULL
49 00175'143010          STB      2,0
50 00176'020411          LDA      0,.TP
51 00177'101220          MOVZR    0,0      ;SET WORD ADDRESS
52 00200'040404          STA      0,B3.1
53
54 00201'006001$          JSR      0,FCAL
55 00202'077777          KFILL
56 00203'000002          2
57 00204'000000 B3.1:    0
58 00205'000402          IER
59 00206'000402          JMP      B4
60
61 00207'002612".TP:    TP*2
62

```

```

0004 STORA
01
02 00210'006017 B4: .SYSTEM ;ALL BRANCHES JOIN HERE
03 00211'021006 .GDAY ;SET DTG FOR HEADER
04 00212'002001- JMP @.ERR
05 00213'034553 LDA 3,.HDG
06 00214'045403 STA 1,3,3 ;MONTH
07 00215'041404 STA 0,4,3 ;DAY
08 00216'006017 .SYSTEM
09 00217'021003 .GTOD
10 00220'002001- JMP @.ERR
11 00221'034545 LDA 3,.HDG
12 00222'051405 STA 2,5,3 ;HOUR
13 00223'045406 STA 1,6,3 ;MINUTES
14 00224'020703 LDA 0,.DP ;ADD DIRECTORY NAME TO SYS.DR FILENAME
15 00225'024563 LDA 1,.FN
16 00226'004524 JSR MDN
17 00227'020712 LDA 0,.SD
18 00230'030545 LDA 2,C7
19 00231'006003- JSR @.MCHR
20
21 00232'006017 B8: .SYSTEM ;OPEN A CHANNEL TO SYS.DR
22 00233'021052 .GCHN
23 00234'002001- JMP @.ERR
24 00235'020553 LDA 0,.FN
25 00236'126400 SUB 1,1
26 00237'006017 .SYSTEM
27 00240'014077 .OPEN 77
28 00241'002001- JMP @.ERR
29 00242'050543 STA 2,SDC
30 00243'102400 SUB 0,0
31 00244'040525 STA 0,BLK
32
33 00245'020542 B9: LDA 0,.DBUF ;READ A BLOCK OF SYS.DR
34 00246'024523 LDA 1,BLK
35 00247'030536 LDA 2,SDC
36 00250'034523 LDA 3,C400 ;BLOCK COUNT
37 00251'173000 ADD 3,2
38 00252'006017 .SYSTEM
39 00253'005477 .RDB 77
40 00254'002514 JMP @.B17 ;POSSIBLE EOF
41 00255'010514 ISZ BLK
42 00256'030531 LDA 2,.DBUF
43 00257'021000 LDA 0,0,2
44 00260'040523 STA 0,NF ;NUMBER OF FILENAMES
45 00261'101005 MOV 0,0,SNR
46 00262'000763 JMP B9 ;NO FILENAMES IN THIS BLOCK
47 00263'151400 INC 2,2
48
49 00264'021000 B10: LDA 0,0,2
50 00265'101004 MOV 0,0,SZR
51 00266'000404 JMP B10.1
52 00267'020503 LDA 0,C22 ;SKIP DELETED FILENAMES
53 00270'113000 ADD 0,2
54 00271'000773 JMP B10
55
56 00272'050512 B10.1: STA 2,S2
57 00273'021005 LDA 0,5,2 ;EXTENSION
58 00274'101004 MOV 0,0,SZR
59 00275'002472 JMP @.B16 ;SKIP IF NOT ZERO
60 00276'151120 MOVZL 2,2 ;BYTE POINTER TO FILENAME IN SYS.DR
61 00277'050511 STA 2,.FN
62 00300'020476 LDA 0,D10 ;MAX. 10 CHARACTERS

```

```

0005 STORA
01 00301'040000- STA 0,COUNT
02 B11:
03 00302'142710 LDB 2,0 ;COUNT FILENAME CHARACTERS
04 00303'101005 MOV 0,0,SNR
05 00304'000405 JMP B11.1
06 00305'151400 INC 2,2
07 00306'014000- DSZ COUNT
08 00307'000773 JMP B11
09 00310'002457 JMP 0,B16 ;10 CHARACTERS---SKIP
10 B11.1:
11 00311'020463 LDA 0,C4
12 00312'024000- LDA 1,COUNT ;NCHAR=10-COUNT
13 00313'105010 SGT 0,1
14 00314'002453 JMP 0,B16 ;SKIP
15 00315'020004- LDA 0,ALL ;CHECK FOR ALL FILES
16 00316'101004 MOV 0,0,SZR
17 00317'000474 JMP B12 ;USE
18 00320'024005- LDA 1,GFLAG
19 00321'030007- LDA 2,C6 ;CHECK 6 CHARACTERS FOR ALPHA/NUMERICS
20 00322'125004 MOV 1,1,SZR
21 00323'151400 INC 2,2 ;CHECK 7 CHARACTERS FOR GRAPHICS
22 00324'020464 LDA 0,.FN
23 00325'024662 LDA 1,.TP ;BYTE POINTER TO TEMPLATE
24 00326'004404 JSR MATCH
25 00327'101005 MOV 0,0,SNR
26 00330'002437 JMP 0,B16 ;SKIP
27 00331'000462 JMP B12
28
29 MATCH:
30 00332'054416 STA 3,MA3 ;ENTER WITH AC0 AND AC1
31 00333'050416 STA 2,MC ;CONTAINING BYTE POINTERS TO
32 MA1: ;TEXT STRINGS
33 00334'112710 LDB 0,2
34 00335'136710 LDB 1,3
35 00336'156405 SUB 2,3,SNR ;ENTER WITH AC2 CONTAINING
36 00337'000403 JMP MA2 ;NO. OF CHARACTERS TO COMPARE
37 00340'102400 SUB 0,0
38 00341'002407 JMP 0,MA3 ;RETURN WITH (AC0)=-1 IF
39 MA2: ;STRINGS MATCH, OTHERWISE
40 00342'101400 INC 0,0 ;(AC0)=0
41 00343'125400 INC 1,1
42 00344'014405 DSZ MC
43 00345'000767 JMP MA1
44 00346'102000 ADC 0,0
45 00347'002401 JMP 0,MA3
46 00350'000000 MA3: 0
47 00351'000000 MC: 0
48
49 MDN:
50 00352'112710 LDB 0,2 ;MOVE DIRECTORY NAME
51 00353'133010 STB 1,2 ;TO BYTE ADDRESS IN AC1
52 00354'151005 MOV 2,2,SNR
53 00355'000404 JMP MDN2
54 00356'101400 INC 0,0
55 00357'125400 INC 1,1
56 00360'000772 JMP MDN
57 MDN2:
58 00361'030404 LDA 2,MDN72
59 00362'133010 STB 1,2
60 00363'125400 INC 1,1
61 00364'001400 JMP 0,3
00365'000072 MDN72: 72

```

```

0006 STORA
01
02 00366'001274'.HDG: HDG
03 00367'000527'.B16: B16
04 00370'000535'.B17: B17
05 00371'000000 BLK: 0 ;BLOCK NO. OF SYS.DR
06 00372'000022 C22: 22
07 00373'000400 C400: 400
08 00374'000004 C4: 4
09 00375'000007 C7: 7
10 00376'000012 D10: 10.
11 00377'000022 D18: 18.
12 00400'000011 D9: 9.
13 00401'000000 FCH: 0 ;CHANNEL NO. OF DISK FILE
14 00402'000000 IER: 0 ;ERROR RETURN FROM FCALLS
15 00403'000000 NF: 0 ;NO. OF FILENAMES PER BLOCK
16 00404'000000 S2: 0 ;STORAGE FOR AC2
17 00405'000000 SDC: 0 ;SYS.DR CHANNEL
18 00406'001444".BUF: BUF*2
19 00407'000634".DBUF: DBUF
20 00410'002514".FN: FN*2
21 00411'002540".OFN: OFN*2
22 00412'003364".ST: ST*2
23
24 B12:
25 00413'102000 ADC 0,0
26 00414'040006- STA 0,FLAG ;SET PRODUCT FLAG TO TRUE
27 00415'122470 ELDA 0,.DP
28 077511
29 00417'024772 LDA 1,.OFN
30 00420'004732 JSR MDN ;MOVE THE DIRECTORY NAME
31 00421'020767 LDA 0,.FN ;AND CREATE OUTPUT FILENAME
32 00422'030754 LDA 2,D10
33 00423'006003- JSR @.MCHR
34 00424'006017 .SYSTEM ;READ THE FILE
35 00425'021052 .GCHN
36 00426'002001- JMP @.ERR
37 00427'020762 LDA 0,.OFN
38 00430'126400 SUB 1,1
39 00431'006017 .SYSTEM
40 00432'014077 .OPEN 77
41 00433'002001- JMP @.ERR
42 00434'020752 LDA 0,.BUF
43 00435'024742 LDA 1,D18 ;READ 18 BYTES
44 00436'006017 .SYSTEM
45 00437'015077 .RDS 77
46 00440'002001- JMP @.ERR
47 00441'050740 STA 2,FCH ;FILE CHANNEL
48 00442'030734 LDA 2,D10
49 00443'024000- LDA 1,COUNT
50 00444'132400 SUB 1,2 ;CHARACTERS IN FILENAME
51 00445'020743 LDA 0,.FN
52 00446'024740 LDA 1,.BUF ;HEADER ON DISK FILE
53 00447'004663 JSR MATCH
54 00450'101005 MOV 0,0,SNR
55 00451'000456 JMP B16 ;SKIP
56 00452'034734 LDA 3,.BUF ;MOVE NEW HEADING TO FILE
57 00453'020725 LDA 0,D9
58 00454'117000 ADD 0,3
59 00455'030711 LDA 2,.HDG
60 00456'040000- STA 0,COUNT
61 B13:
62 00457'021000 LDA 0,0,2

```

```

0007 STORA
01 00460'163010      STB      3,0
02 00461'151400      INC      2,2
03 00462'175400      INC      3,3
04 00463'014000-     DSZ      COUNT
05 00464'000773      JMP      B13
06
07 00465'102400      SUB      0,0      ;REWIND THE FILE
08 00466'126400      SUB      1,1
09 00467'030712      LDA      2,FCH
10 00470'006017      .SYSTEM
11 00471'023477      .SPOS      77
12 00472'002001-     JMP      @.ERR
13 00473'020713      LDA      0,.BUF      ;AND WRITE THE NEW HEADING
14 00474'024703      LDA      1,D18
15 00475'006017      .SYSTEM
16 00476'016477      .WRS      77
17 00477'002001-     JMP      @.ERR
18 00500'006017      .SYSTEM
19 00501'014477      .CLOSE      77
20 00502'002001-     JMP      @.ERR
21 00503'020706      LDA      0,.OFN
22 00504'101220      MOVZR      0,0
23 00505'040404      STA      0,B14
24
25 00506'006001$     JSR      @.FCAL      ;STORE THE FILE
26 00507'077777      USTOR      2
27 00510'000002      0
28 00511'000000 B14: 0
29 00512'000402'     IER
30 00513'030667      LDA      2,IER
31 00514'150110      SBI      3,2
32 00515'014665      DSZ      IER
33 00516'002001-     JMP      @.ERR
34 00517'030667      LDA      2,.BUF      ;TYPE PRODUCT KEY
35 00520'034660      LDA      3,D9
36 00521'157000      ADD      2,3
37 00522'102400      SUB      0,0      ;STORE NULL
38 00523'163010      STB      3,0      ; AT END OF PRODUCT KEY
39 00524'006002-     JSR      @.SPCH
40 00525'030665      LDA      2,.ST
41 00526'006002-     JSR      @.SPCH      ;" STORED"
42
43 00527'030655      LDA      2,S2      B16:
44 00530'020642      LDA      0,C22
45 00531'113000      ADD      0,2
46 00532'014651      DSZ      NF
47 00533'002414      JMP      @.B10      ;FETCH NEXT FILENAME
48 00534'002412      JMP      @.B9      ;READ NEXT BLOCK
49
50 00535'020007-     LDA      0,C6      B17:
51 00536'142404      SUB      2,0,SZR
52 00537'002001-     JMP      @.ERR
53 00540'030645      LDA      2,SDC
54 00541'006017      .SYSTEM
55 00542'014477      .CLOSE      77
56 00543'002001-     JMP      @.ERR
57 00544'002401      JMP      @.B1
58 00545'000046' .B1: B1
59 00546'000245' .B9: B9
60 00547'000264' .B10: B10
61
62      B18:

```

```

0008 STORA
01 00550'030405      LDA      2, .M11      ;BUTTONUP
02 00551'006002-      JSR       @.SPCH
03 00552'006017      .SYSTM
04 00553'004400      .RTN
05 00554'000776      JMP       -2
06 00555'003336" .M11: M11*2
07
08
09 00556'050000-      STA      2,COUNT
10 00557'030406      LDA      2, .M10      ;SYSTEM ERROR
11 00560'006002-      JSR       @.SPCH
12 00561'030000-      LDA      2,COUNT
13 00562'006017      .SYSTM
14 00563'006400      .ERTN
15 00564'000776      JMP       -2
16 00565'003320" .M10: M10*2
17
18
19 00566'050410      STA      2,MCHC      ;MOVE (AC2) CHARACTERS FROM
20 00567'112710      LDB       0,2      ;BYTE ADDRESS IN AC0
21 00570'133010      STB       1,2      ;TO BYTE ADDRESS IN AC1
22 00571'101400      INC       0,0
23 00572'125400      INC       1,1
24 00573'014403      DSZ       MCHC
25 00574'000773      JMP       MCHR+1
26 00575'001400      JMP       0,3
27 00576'000000 MCHR: 0
28
29
30 00577'054422      STA      3,SAV3      ;$TTO OUTPUT ROUTINE
31 00600'142710      LDB       2,0
32 00601'101005      MOV       0,0,SNR
33 00602'002417      JMP       @SAV3
34 00603'006017      .SYSTM
35 00604'010000      .PCHAR
36 00605'000401      JMP       .+1
37 00606'151400      INC       2,2
38 00607'024410      LDA      1,S15
39 00610'106404      SUB       0,1,SZR
40 00611'000767      JMP       SPCHR+1
41 00612'020406      LDA      0,S12
42 00613'006017      .SYSTM
43 00614'010000      .PCHAR
44 00615'000401      JMP       .+1
45 00616'002403      JMP       @SAV3
46 00617'000015 S15: 15
47 00620'000012 S12: 12
48 00621'000000 SAV3: 0
49
50 00622'000012 BUF: .BLK 10.      ;HEADER BYTES FROM FILE
51 00634'000400 DBUF: .BLK 256.    ;SYS.DR BLOCK
52 01234'000012 DP: .BLK 10.      ;SOURCE DIRECTORY
53 01246'000012 FN: .BLK 10.      ;DP*:SYS.DR OR PRODUCT FILENAME
54 01260'000014 OFN: .BLK 12.     ;OUTPUT FILENAME
55 01274'000060 HDG: 60      ;ADDRESSEE
56 01275'000060      60
57 01276'000060      60
58 01277'000000      0      ;MONTH
59 01300'000000      0      ;DAY
60 01301'000000      0      ;HOUR
61 01302'000000      0      ;MINUTES
62 01303'000005      5      ;PRIORITY

```

```

0009 STORA
01 01304'000000      0      ;TYPE
02 01305'000005 TP:   .BLK    5      ;OPERATOR TEMPLATE
03 01312'040514 A:    .TXT     "ALL"
04 01314'046517 M1:    .TXT     "MOVE AFOS PRODUCTS FROM A DISK TO THE DATABASE<15>"
05 01344'042516 M2:    .TXT     "ENTER SOURCE DIRECTORY NAME (E.G., DP3)      "
06 01373'040506 M4:    .TXT     "AFTER A * PROMPT ENTER 1 CHARACTER FOR GRAPHICS,<15>"
07 01424'031440 M5:    .TXT     "3 CHARACTERS FOR A LOCAL NNN, 6 CHARACTERS FOR A REMOTE
08 01457'042440 CCCNNN,<15>"
09 01465'023501 M6:    .TXT     "'ALL' FOR ALL VALID FILES, OR [RETURN] TO TERMINATE
10 01516'042440 PROGRAM<15>"
11 01524'047117 M7:    .TXT     "NO FILES MATCH SPECIFIER<15>"
12 01541'044516 M9:    .TXT     "INPUT ERROR<15>"
13 01550'051531 M10:   .TXT     "SYSTEM ERROR<15>"
14 01557'045117 M11:   .TXT     "JOB COMPLETE<15>"
15 01566'051531 SD:    .TXT     "SYS.DR"
16 01572'020123 ST:    .TXT     "STORED<15>"
17 01577'022124 TTF:   .TXT     "$TTI"
18 01602'047115 NMC:   .TXT     "NMCGPH"
19      000000      .NOLOC  0
20      .END

```

***00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0010 STORA

A	0001312'	3/04	9/03					
ALL	0000004-	1/34	2/42	3/39	5/15			
B1	0000046'	2/17	7/58					
B10	000264'	4/48	4/54	7/60				
B10.1	000272'	4/51	4/55					
B11	000302'	5/02	5/08					
B11.1	000311'	5/05	5/10					
B12	000413'	5/17	5/27	6/24				
B13	000457'	6/61	7/05					
B14	000511'	7/23	7/28					
B16	000527'	6/03	6/55	7/42				
B17	000535'	6/04	7/49					
B18	000550'	3/05	7/62					
B1.1	000055'	2/16	2/20	2/25	2/54			
B1.2	000112'	2/07	2/55					
B2	000144'	2/45	3/22					
B3	000157'	2/48	3/34					
B3.1	000204'	3/52	3/57					
B4	000210'	2/51	3/33	3/41	3/59	4/01		
B8	000232'	4/20						
B9	000245'	4/32	4/46	7/59				
BLK	000371'	4/31	4/34	4/41	6/05			
BUF	000622'	6/18	8/50					
C15	000117'	2/62						
C22	000372'	4/52	6/06	7/44				
C3	000120'	2/45	3/01	3/37	3/43			
C4	000374'	5/11	6/08					
C400	000373'	4/36	6/07					
C45	000116'	2/56	2/61					
C6	000007-	1/37	2/49	3/23	5/19	7/50		
C60	000121'	3/02	3/45					
C7	000375'	4/18	6/09					
COUNT	000000-	1/30	5/01	5/07	5/12	6/49	6/60	7/04
		8/09	8/12					
D10	000376'	4/62	6/10	6/32	6/48			
D18	000377'	6/11	6/43	7/14				
D9	000400'	6/12	6/57	7/35				
DBUF	000634'	6/19	8/51					
DP	001234'	3/00	8/52					
ERR	000556'	1/31	8/08					
FCH	000401'	6/13	6/47	7/09				
FLAG	000006-	1/36	2/18	2/27	6/26			
FN	001246'	6/20	8/53					
GFLAG	000005-	1/35	2/41	3/32	5/18			
HDG	001274'	6/02	8/55					
IER	000402'	3/50	6/14	7/29	7/30	7/32		
KFILL	000202' XN	1/04	3/55					
M1	001314'	3/09	9/04					
M10	001550'	8/16	9/13					
M11	001557'	8/06	9/14					
M2	001344'	3/10	9/05					
M4	001373'	3/11	9/06					
M5	001424'	3/12	9/07					
M6	001465'	3/13	9/09					
M7	001524'	3/14	9/11					
M9	001541'	3/15	9/12					
MA1	000334'	5/32	5/43					
MA2	000342'	5/36	5/39					
MA3	000350'	5/30	5/38	5/45	5/46			
MATCH	000332'	3/38	5/24	5/29	6/53			
MC	000351'	5/31	5/42	5/47				
MCHC	000576'	8/19	8/24	8/27				

0011 STORA

MCHR	000566'	1/33	8/18	8/25				
MDN	000352'	4/16	5/49	5/56	6/30			
MDN2	000361'	5/53	5/57					
MDN72	000365'	5/58	5/62					
NF	000403'	4/44	6/15	7/46				
NMC	001602'	3/20	9/18					
OFN	001260'	6/21	8/54					
S12	000620'	8/41	8/47					
S15	000617'	8/38	8/46					
S2	000404'	4/56	6/16	7/43				
SAV3	000621'	8/30	8/33	8/45	8/48			
SD	001566'	3/18	9/15					
SDC	000405'	4/29	4/35	6/17	7/53			
SPCHR	000577'	1/32	8/29	8/40				
ST	001572'	6/22	9/16					
TP	001305'	3/61	9/02					
TTC	000122'	1/49	1/58	2/33	3/03			
TTF	001577'	3/19	9/17					
USTOR	000507' XN	1/04	7/26					
.A	000123'	3/04	3/36					
.B1	000545'	7/57	7/58					
.B10	000547'	7/47	7/60					
.B16	000367'	4/59	5/09	5/14	5/26	6/03		
.B17	000370'	4/40	6/04					
.B18	000124'	2/39	3/05					
.B9	000546'	7/48	7/59					
.BUF	000406'	6/18	6/42	6/52	6/56	7/13	7/34	
.CR	000125'	1/53	2/08	2/21	2/28	3/06		
.DBUF	000407'	4/33	4/42	6/19				
.DP	000127'	1/57	3/08	4/14	6/27			
.ERR	000001-	1/31	1/43	1/48	1/61	2/36	2/58	4/04
		4/10	4/23	4/28	6/36	6/41	6/46	7/12
		7/17	7/20	7/33	7/52	7/56		
.FCAL	000001\$ XD	1/03	3/54	7/25				
.FN	000410'	4/15	4/24	4/61	5/22	6/20	6/31	6/51
.HDG	000366'	4/05	4/11	6/02	6/59			
.I	077777' XN	1/04						
.M1	000130'	1/51	3/09					
.M10	000565'	8/10	8/16					
.M11	000555'	8/01	8/06					
.M2	000131'	1/55	3/10					
.M4	000132'	2/10	3/11					
.M5	000133'	2/12	3/12					
.M6	000134'	2/14	3/13					
.M7	000135'	2/23	3/14					
.M9	000136'	2/52	3/15					
.MAIN	000001' EN	1/02	1/40					
.MCHR	000003-	1/33	3/30	4/19	6/33			
.NMC	000143'	3/20	3/28					
.OFN	000411'	6/21	6/29	6/37	7/21			
.PMT	000137'	2/30	3/16					
.SD	000141'	3/18	4/17					
.SPCH	000002-	1/32	1/52	1/54	1/56	2/09	2/11	2/13
		2/15	2/22	2/24	2/29	2/31	2/53	7/39
		7/41	8/02	8/11				
.ST	000412'	6/22	7/40					
.TP	000207'	2/32	3/24	3/29	3/35	3/42	3/50	3/61
		5/23						
.TTF	000142'	1/44	3/19					

APPENDIX 5

```

0001 USTOR MACRO REV 06.00          08:18:58 06/20/82
      .TITL    USTOR                ;UNBUFFERED AFOS STORE ROUTINE
02      .ENT    USTOR
03      .EXTD   .CPYL, .FRET, .FCAL
04      .EXTN   FSTOR
05      ;
06      ;      SUBROUTINE USTORE(FILENAME, IER)
07      ;
08      ;      STORE A FILE IN THE AFOS DATABASE AND WAIT FOR COMPLETION BEFORE
09      ;      RETURNING TO THE CALLING PROGRAM.
10      ;
11      ;      IER IS FORTRAN SYSTEM CODE
12      ;
13      ;      PROGRAMMER:  WARREN E. SUNKEL, WSFO TOPEKA
14      ;
15      177600      FZD      =-200
16      177611      FN      =-167
17      177612      IER     =FN+1
18      177613      UFD     =IER+1
19      177635      .UFD    =UFD+22
20      000025      .FS     =.UFD+1-FN
21
22      .NREL
23 00000'000025      .FS
24      USTOR:
25 00001'006003$      JSR     @.CPYL          ;COPY STACK LINKAGE
26 00002'054457      STA     3,FSP          ;AND SAVE FRAME POINTER
27 00003'021611      LDA     0,FN,3
28 00004'101120      MOVZL   0,0
29 00005'000402      JMP     .+2
30 00006'177613      UFD
31 00007'024777      LDA     1,.-1
32 00010'167000      ADD     3,1            ;COMPUTE BASE ADDRESS FOR UFD
33 00011'045635      STA     1,.UFD,3
34 00012'006017      .SYSTEM              ;ENSURE FILE IS CLOSED
35 00013'021014      .RSTAT
36 00014'000441      JMP     ERR
37 00015'030445      LDA     2,ERCD          ;FILE IN USE
38 00016'034443      LDA     3,FSP
39 00017'021633      LDA     0,UFD+20,3
40 00020'101004      MOV     0,0,SZR
41 00021'000434      JMP     ERR
42
43 00022'006001$      JSR     @.FCAL          ;START THE STORAGE PROCESS
44 00023'077777      FSTOR
45 00024'000003      3
46 00025'100011      FN-FZD+1B0
47 00026'000063'      C0
48 00027'100012      IER-FZD+1B0
49
50 00030'034431      LDA     3,FSP
51 00031'021611      LDA     0,FN,3
52 00032'101120      MOVZL   0,0
53 00033'025635      LDA     1,.UFD,3
54      U1:
55 00034'006017      .SYSTEM
56 00035'021014      .RSTAT
57 00036'000417      JMP     ERR
58 00037'034422      LDA     3,FSP
59 00040'031633      LDA     2,UFD+20,3
60 00041'151005      MOV     2,2,SNR
61 00042'000772      JMP     U1            ;LOOP UNTIL FILE OPENS
62      U2:

```

```

0002 USTOR
01 00043'006017 .SYSTEM
02 00044'021014 .RSTAT
03 00045'000410 JMP ERR
04 00046'034413 LDA 3,FSP
05 00047'031633 LDA 2,UFD+20,3
06 00050'151004 MOV 2,2,SZR
07 00051'000772 JMP U2 ;LOOP UNTIL FILE CLOSES
08 00052'102520 SUBZL 0,0 ;CLEAR ERROR CODE
09 00053'041612 STA 0,IER,3
10 00054'006002$ JSR 0,FRET ;RETURN TO CALLING PROGRAM
11 ERR:
12 00055'150010 ADI 3,2 ;ADD FORTRAN BIAS
13 00056'034403 LDA 3,FSP
14 00057'053612 STA 2,0,IER,3
15 00060'006002$ JSR 0,FRET
16
17 00061'000000 FSP: 0
18 00062'000060 ERCD: ERFIU
19 00063'000000 C0: 0
20 .END

```

***00000 TOTAL ERRORS, 00000 PASS 1 ERRORS

0003 USTOR

C0	000063'	1/47	2/19					
ERCD	000062'	1/37	2/18					
ERR	000055'	1/36	1/41	1/57	2/03	2/11		
FN	177611	1/16	1/17	1/20	1/27	1/46	1/51	
FSP	000061'	1/26	1/38	1/50	1/58	2/04	2/13	2/17
FSTOR	000023' XN	1/04	1/44					
FZD	177600	1/15	1/46	1/48				
IER	177612	1/17	1/18	1/48	2/09	2/14		
U1	000034'	1/54	1/61					
U2	000043'	1/62	2/07					
UFD	177613	1/18	1/19	1/30	1/39	1/59	2/05	
USTOR	000001' EN	1/02	1/24					
.CPYL	000003\$ XD	1/03	1/25					
.FCAL	000001\$ XD	1/03	1/43					
.FRET	000002\$ XD	1/03	2/10	2/15				
.FS	000025	1/20	1/23					
.UFD	177635	1/19	1/20	1/33	1/53			



NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

NOAA, the *National Oceanic and Atmospheric Administration*, was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth, and to assess the socioeconomic impact of natural and technological changes in the environment.

The six Major Line Components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

TECHNICAL REPORTS—Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

CONTRACT AND GRANT REPORTS—Reports prepared by contractors or grantees under NOAA sponsorship.

TECHNICAL SERVICE PUBLICATIONS—These are publications containing data, observations, instructions, etc. A partial listing: Data serials; Prediction and outlook periodicals; Technical manuals, training papers, planning reports, and information serials; and Miscellaneous technical publications.

ATLAS—Analysed data generally presented in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.



Information on availability of NOAA publications can be obtained from:

**ENVIRONMENTAL SCIENCE INFORMATION CENTER
ENVIRONMENTAL DATA SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE**

**3300 Whitehaven Street, N.W.
Washington, D.C. 20235**