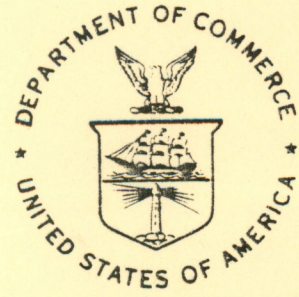


H
QC
874.3
U68
no.19

NOAA Western Region Computer Programs and
Problems NWS WRCP - No. 19



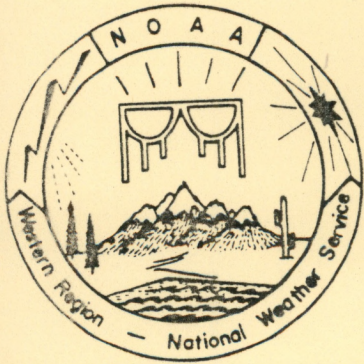
DATAKEYØ REPAIR PROGRAM

National Weather Service Western Region
Salt Lake City, Utah
August 1980

**U.S. DEPARTMENT OF
COMMERCE**

National Oceanic and
Atmospheric Administration

National Weather
Service



This Western Region publication series is considered as a subset of our Technical Memorandum series. This series will be devoted exclusively to the exchange of information on and documentation of computer programs and related subjects. This series was initiated because it did not seem appropriate to publish computer program papers as Technical Memoranda; yet, we wanted to share this type of information with all Western Region forecasters in a systematic way. Another reason was our concern that in the developing AFOS-era there will be unnecessary and wasteful duplication of effort in writing computer programs in National Weather Service (NWS). Documentation and exchange of ideas and programs envisioned in this series hopefully will reduce such duplication. We also believe that by publishing the programming work of our forecasters, we will stimulate others to use these programs or develop their own programs to take advantage of the computing capabilities AFOS makes available.

We solicit computer-oriented papers and computer programs from forecasters for us to publish in this series. Simple and short programs should not be prejudged as unsuitable.

The great potential of the AFOS-era is strongly related to local computer facilities permitting meteorologists to practice in a more scientific environment. It is our hope that this new series will help in developing this potential into reality.

NOAA Western Region Computer Programs and Problems NWS WRCP

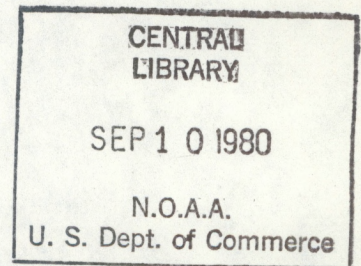
- 1 Standard Format for Computer Series. June 1979.
- 2 AFOS Crop and Soil Information Report Program. Ken Mielke, July 1979.
- 3 Decoder for Significant Level Transmission of Raobs. John Jannuzzi, August 1979.
- 4 Precipitable Water Estimate. Elizabeth Morse, October 1979.
- 5 Utah Recreational Temperature Program. Kenneth M. Labas, November 1979.
- 6 Normal Maximum/Minimum Temperature Program for Montana. Kenneth Mielke, December 1979.
- 7 Plotting of Ocean Wave Energy Spectral Data. John R. Zimmerman, December 1979.
- 8 Raob Plot and Analysis Routines. John Jannuzzi, January 1980.
- 9 The SWAB Program. Morris S. Webb, Jr., April 1980. (PB 80-196041)
- 10 Flash-Flood Procedures. Donald P. Laurine and Ralph C. Hatch, April 1980. (PB 80-198658)
- 11 Program to Forecast Probability of Summer Stratus in Seattle Using the Durst Objective Method. John Zimmerman, May 1980.
- 12 Probability of Sequences of Wet and Dry Days. Hazen H. Bedke, June 1980.
- 13 Automated Montana Hourly Weather Roundup. Joe L. Johnston, July 1980.
- 14 Lightning Activity Levels. Mark A. Mollner, July 1980.
- 15 Two Fortran Applications of Wind-Driven Ekman Water Transport Theory: Upwelling Index and Storm Tide. Kent S. Short, July 1980.
- 16 AFOS System Local Database Save and Rebuild Procedures or A Master Doomsday Program. Brian W. Finke, July 1980.
- 17 AFOS/RDOS Translator Subroutine. Morris S. Webb, Jr., August 1980.
- 18 AFOS Graphics Creation from Fortran. Alexander E. MacDonald, August 1980.

H
GC
874.3
U68
no. 19

NOAA Western Region Computer Programs and Problems NWS WRCP - No. 19

DATAKEYØ REPAIR PROGRAM

Paul D. Tolleson
Scientific Services Division
Western Region Headquarters
Salt Lake City, Utah
August 1980



UNITED STATES
DEPARTMENT OF COMMERCE
Philip M. Klutznick, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

National Weather
Service
Richard E. Hallgren, Director



CONTENTS

	<u>Page</u>
I. General Information	1
II. Application	1
III. Procedures	2

DATAKEYØ REPAIR PROGRAM

Paul D. Tolleson
Summer Meteorological Aid
Scientific Services Division
Western Region Headquarters
Salt Lake City, Utah

I. General Information

A. Summary:

The DATAKEYØ repair program DKØRPR is a general utility program that is beneficial in maintaining the integrity of DATAKEYØ. The program detects clobbered keys in DATAKEYØ in AFOS by comparing the keyname of a key in DATAKEYØ with the corresponding key in a backup copy of DATAKEYØ on floppy disk. The wish list is not compared since it is continually updated. During execution, the user must direct the program to correct or to simply list any discrepancies found. In addition, starting and ending record numbers can also be specified by the user to define the bounds of the search.

B. Environment:

This program has been written in DATA GENERAL FORTRAN IV specifically for the ECLIPSE used in AFOS. This program must be run on the background partition since the background dasher (\$TTI) has been specified as the input device.

II. Application

A. Complete Program Description:

The purpose of the DATAKEYØ repair program is to provide a rather simple procedure to detect discrepancies in the very important DATAKEYØ file on the DPØ disk in AFOS. Through comparison with a sufficiently recent backup copy of DATAKEYØ on floppy disk, the skeleton program finds changes in the keynames in DATAKEYØ. However, a number of important features at the disposal of the user can extend the scope of the program.

First, the user must specify whether he wants to have the program correct the discrepancies found or whether he wants only to have the discrepancies listed. In either case, the program prints the octal position of any discrepancy from the beginning of the DATAKEYØ file. Therefore, with this option the user may correct or otherwise change DATAKEYØ.

Second, the user must specify the starting and ending records for the search in DATAKEYØ. This can greatly shorten the run time of the program. Suppose that MODIFY has been run on DATAKEYØ and that it has found a discrepancy in DATAKEYØ which it cannot correct; when this occurs, the number of the record where the discrepancy was found is located in the MODIFY.TX file. Using this information, the user may then run the repair program to correct only that one record.

Regardless of the specified ending record number, the search will terminate at the beginning of the wish list. The program contains this feature because users often change the wish list, not wanting it to be corrected.

Any special (non-DATA GENERAL FORTRAN IV) routines used come from UTIL.LB (the utility library). The following is a list of all UTIL.LB routines found in the program.

CALL GCHN	CALL RDL
CALL ERROR	CALL KLOSE
CALL OPENN	CALL RDS
CALL OPENR	CALL RDB
CALL SPCHR	CALL WRB

B. Machine Requirements:

DKØRPR runs in 10K of memory. The save file requires 12,800 bytes of storage. Run time is approximately 1-1/2 minutes, but will increase as more discrepancies are found because of the time needed for the dasher to print output.

C. Database:

This program accesses a number of files in performing its task. The most important file is DATAKEYØ. DATAKEYØ consists of numerous RDOS blocks which are arranged in groups of four (called a record). Each record consists of 1024 words which are arranged in groups of 15. Each group of 15 words is called a key. The first 4-1/2 words (9 bytes) of each key is the keyname; this is precisely what is compared in the program. Following the main section of DATAKEYØ, one encounters the wish list. This is not compared.

The user needs to realize the internal labeling system present in DATAKEYØ. Words are labeled from the start of the file beginning with 'Ø' (not '1'). Records are similarly labeled.

The second important file accessed is the backup copy of DATAKEYØ on a floppy disk. This file must exist before the program is executed and while DATAKEYØ is known to be accurate. This backup file must be contiguous or random, since the CALL RDB routine will not read sequential files. This backup must be replaced after any PILEEDIT.

The third file accessed is also on DPØ. It is SKEL. The importance of SKEL is that word 162 (octal) provides the program with the beginning of the wish list and, therefore, the criterion to terminate execution.

III. Procedures

A. Initiation of Program:

Before initiating the program, the user must perform some preliminary tasks. First, he must be sure that an accurate backup contiguous or random copy of DATAKEYØ exists on floppy disk. Next, this floppy disk must be initialized into the system (i.e., INIT DP3).

The user invokes the program by simply entering (from the directory in which the program resides) the following at the dasher:

```
DKØRPR.
```

B. Input Required:

The program accesses three files and reads user-input. The three files needed by the program are DATAKEYØ, the backup copy of DATAKEYØ, and SKEL. The following paragraphs describe the necessary user input.

Once the user initiates the program from the directory on which it resides, the program will print at the dasher:

```
ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY--  
DP?:FILENAME--
```

On the same line the user enters the filename of the backup copy with the floppy drive being used, followed by a carriage return.

Next the program generates the statement:

```
TYPE 'Ø' TO REPLACE DISCREPANT KEYS;TYPE '1' TO CHECK FOR  
DISCREPANCIES ONLY
```

This requires the user to enter 'Ø' or '1', followed by a carriage return. A '1' will make the program only list discrepancies found; a 'Ø' will allow the program to correct any discrepancies found, as well as to list them.

After the above has been executed, the dasher prints:

```
STARTING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)
```

The user now enters on the same line (followed by a carriage return) the number of the record where he wants the search to begin. A 'Ø' will start the search at the beginning of DATAKEYØ.

Finally, the message:

```
ENDING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)
```

appears at the dasher. The number of the ending record of the search must be entered on the same line, followed by a carriage return. This final record is not searched for discrepancies. A 'Ø' will automatically cause the program to terminate upon reaching the wish list. Note that the wish list is never compared.

Below is a listing of sample user input necessary for the complete execution of this program:

```
DKØRPR.SV
ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY--
DP?:FILENAME--DP3:DATAKEYØ
TYPE 'Ø' TO REPLACE DISCREPANT KEYS;TYPE '1' TO CHECK
FOR DISCREPANCIES ONLYØ
STARTING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
ENDING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
```

NOTE: All underlined entries are user input.

C. Output:

This program may generate any number of output statements, depending on whether any discrepancies exist, as well as on the user input described under INPUT REQUIRED. The three possible cases will be discussed.

- 1) If under any circumstances no discrepancies are found, the dasher will print:

```
PROGRAM FINISHED
```

An R prompt will appear shortly indicating that the program has terminated. The following is example output.

```
DKØRPR.SV
ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY--
DP?:FILENAME--DP3:DATAKEYØ
TYPE 'Ø' TO REPLACE DISCREPANT KEYS; TYPE '1' TO CHECK
FOR DISCREPANCIES ONLY1
STARTING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
ENDING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
PROGRAM FINISHED
R
```

NOTE: All underlined entries are user input.

- 2) If the user has directed the program to only list discrepancies, the dasher will type

BAD KEY; WORD NUMBER=?? OCTAL

(whenever a discrepancy is found). The word number gives the position from the beginning of DATAKEYØ of the discrepant word in octal. Next, the clobbered keyname and the correct keyname are printed. After this is done for each discrepancy, the statement:

PROGRAM FINISHED

followed by an R prompt will indicate that the program has terminated. The following is example output.

```
DKØRPR.SV
ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY--
DP?:FILENAME--DP3:DATAKEYØ
TYPE 'Ø' TO REPLACE DISCREPANT KEYS; TYPE '1' TO CHECK
FOR DISCREPANCIES ONLY
STARTING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
ENDING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)Ø
BAD KEY; WORD NUMBER=          4ØØ. OCTAL
ABLLIRHHS
BAD KEY; WORD NUMBER=          4Ø2. OCTAL
ABQPIRTCS
BAD KEY; WORD NUMBER=          512. OCTAL
TTSPIRDIK
BISPIRDIK
BAD KEY; WORD NUMBER=          626. OCTAL
BIYYIRMIB
BISPIRMIB
BAD KEY; WORD NUMBER=          762. OCTAL
BISSAØ77K
BISSAØGFK
BAD KEY; WORD NUMBER=        1Ø342. OCTAL
BOIIAGBOI
BOIUAGBOI
BAD KEY; WORD NUMBER=        1ØØØØ1. OCTAL
PD44WWPDX
PDXFWWPDX
BAD KEY; WORD NUMBER=        1ØØØ56. OCTAL
PDOOLWEUG
PDXGLWEUG
BAD KEY; WORD NUMBER=        15Ø227. OCTAL
SETTR5OLM
SEARR5OLM
BAD KEY; WORD NUMBER=        155244. OCTAL
SEUUWB359
SEATWB359
PROGRAM FINISHED
R
```

NOTE: All underlined entries are user input.

- 3) If the user has directed the program to correct discrepancies, then other output will be generated. After all the discrepancies in one record have been found with the output under part 2 being printed for each discrepancy, then the statement

RECORD REPLACED, NUMBER OF DISCREPANCIES IN RECORD=??

will indicate that all discrepancies have been rectified in that record and will denote the number of discrepancies previously listed for that record. This will happen for each record which contains a discrepant key. When the search has finished

PROGRAM FINISHED

appears. An R prompt indicates that the program has terminated. The following is example output.

```
DKØRPR.SV
ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY--
DP?:FILENAME--DP3:DATAKEYØ
TYPE 'Ø' TO REPLACE DISCREPANT KEYS;TYPE '1' TO CHECK FOR
DISCREPANCIES ONLYØ
STARTING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)25
ENDING RECORD?(TYPE 'Ø' FOR SEARCH OF ENTIRE FILE)600
BAD KEY; WORD NUMBER= 1ØØØØ1. OCTAL
PD44WWPDX
PDXFWWPDX
BAD KEY; WORD NUMBER= 1ØØØ56. OCTAL
PDOOLWEUG
PDXGLWEUG
RECORD REPLACED. NUMBER OF DISCREPANCIES IN RECORD= 2

BAD KEY; WORD NUMBER= 15Ø227. OCTAL
SETTR5OLM
SEARR5OLM
RECORD REPLACED. NUMBER OF DISCREPANCIES IN RECORD= 1

BAD KEY; WORD NUMBER= 155244. OCTAL
SEUUWB359
SEATWB359
RECORD REPLACED. NUMBER OF DISCREPANCIES IN RECORD= 1

PROGRAM FINISHED
R
```

NOTE: All underlined entries are user input.

```

C INITIALIZE JBYT,ISRT,IEND
  DIMENSION IBUF(1024),NBUF(1024),PP(8),JBUF(200)
  JBYT=400
  ISRT=0
  IEND=0
C OPEN CHANNEL FOR DATAKEY0
  CALL GCHN(ICHN,IER)
  CALL ERROR(IER,"ERROR:GCHN FOR DATAKEY0")
  CALL OPENN(ICHN,"DP0:DATAKEY0",0,IER1)
  CALL ERROR(IER1,"ERROR:OPENING DATAKEY0")
C OPEN CHANNEL FOR DASHER
  CALL GCHN(ICHN1,IER2)
  CALL ERROR(IER2,"ERROR:GCHN FOR #TTI")
  CALL OPENR(ICHN1,"#TTI",0,IER3)
  CALL ERROR(IER3,"ERROR:OPENING #TTI")
C INPUT FROM DASHER
  CALL SPCHR("ENTER THE FOLLOWING INFORMATION CONCERNING BACKUP COPY
  *--DP?:FILENAME--",IER4)
  CALL ERROR(IER4,"ERROR:TYPING MESSAGE")
  CALL RDL(ICHN1,IBUF,IBYT,IER5)
  CALL ERROR(IER5,"ERROR:READING LINE ON CONSOLE")
C CLOSE CHANNEL TO DASHER
  CALL KLOSE(ICHN1,IER6)
  CALL ERROR(IER6,"ERROR:CLOSING CHN FOR #TTI")
C INPUT "0" IF YOU WANT TO REPLACE DISCREPANT KEYS OR "1" IF YOU
C JUST WANT TO CHECK FOR ERRORS; INPUT STARTING RECORD NUMBER
C AND ENDING RECORD NUMBER.
  ACCEPT "TYPE '0' TO REPLACE DISCREPANT KEYS;",
  *"TYPE '1' TO CHECK FOR DISCREPANCIES ONLY",INDEX
  ACCEPT "STARTING RECORD?(TYPE '0' FOR SEARCH OF ENTIRE FILE)",ISRT
  ACCEPT "ENDING RECORD?(TYPE '0' FOR SEARCH OF ENTIRE FILE)",IEND
C OPEN CHANNEL FOR BACKUP DISKETTE
  CALL GCHN(NCHN,NER)
  CALL ERROR(NER,"ERROR:GCHN FOR BACKUP DISKETTE")
  CALL OPENN(NCHN,IBUF,0,NER1)
  CALL ERROR(NER1,"ERROR:OPENING BACKUP DISKETTE")
C OPEN CHANNEL TO SKEL FILE
  CALL GCHN(ICHN12,IER12)
  CALL ERROR(IER12,"ERROR:GCHN FOR SKEL FILE")
  CALL OPENR(ICHN12,"DP0:SKEL",0,IER13)
  CALL ERROR(IER13,"ERROR:OPENING DP0:SKEL")
C READ SKEL BEYOND WORD 162 OCTAL (I.E., THE WORD WHICH GIVES
C THE BLOCK NUMBER FOR THE BEGINNING OF THE WISH LIST IN
C DATAKEY0).
  CALL RDS(ICHN12,JBUF,JBYT,IER14)
  CALL ERROR(IER14,"ERROR:READING SKEL")
  K=-4
C DENOTE STARTING AND ENDING RECORDS
  IF(ISRT.NE.0) K=ISRT#4-8
  IF(IEND.NE.0) IENDR=IEND#4
  IF(IEND.EQ.0) IENDR=JBUF(115)
C LOOP FOR READING 1 RECORD(=4 RDS BLOCKS) INTO AN ARRAY
  20 K=K+4
C EXIT LOOP IF AT WISH LIST IN DATAKEY0
  IF(K.EQ.JBUF(115).OR.K.EQ.IENDR) GO TO 25
  CALL RDB(ICHN,IBUF,K,4,IER7)
  CALL ERROR(IER7,"ERROR:READING DP0:DATAKEY0")
  CALL RDB(NCHN,NBUF,K,4,IER8)
  CALL ERROR(IER8,"ERROR:READING BACKUP DISKETTE")

```

Program Listing:

```

C SET UP COUNTER FOR ERRORS IN EACH RECORD
  N=0
C LOOP TO CHECK EACH KEY
  DO 21 M=1,1015,15
C SET UP COUNTER FOR ERRORS IN EACH KEY OF A RECORD
  N1=0
C LOOP TO CHECK KEYNAME ONLY. IF NOT CORRECT,
C GO TO STATEMENT 10 UPON COMPARISON
  DO 22 J=1,15
C SINGLE OUT WORDS 0,1,2,3,AND 4 OF A KEY FOR SEPARATE COMPARISON.
  IF(J.GT.5) GO TO 22
C SINGLE OUT WORD 4 FOR SEPARATE COMPARISON SINCE ONLY
C HALF OF THE WORD NEEDS COMPARISON
  IF(J.NE.5) GO TO 90
C SHIFT WORD 4 OF A KEY IN DATAKEY0 SINCE ONLY THE LEFT BYTE OF WORD 4
C (KEYNAME) IS COMPARED. THEN COMPARE. IF NOT EQUAL, GO TO
C STATEMENT 10 FOR CORRECTION.
  I15=IBUF(J+M-1)
  K15=ISHFT(I15,-8)
  I25=NBUF(J+M-1)
  K25=ISHFT(I25,-8)
C COMPARE WORD 4
  IF(K15.NE.K25) GO TO 10
  GO TO 22
C COMPARE WORDS NUMBER 0,1,2,3.
  90 IF(IBUF(J+M-1).NE.NBUF(J+M-1)) GO TO 10
  GO TO 22
C CHECK POSITION OF ERRORS
C CONVERT DECIMAL WORD NUMBER OF CLOBBED WORD TO OCTAL
  10 X1=FLOAT(K)
  X2=FLOAT(J)
  X3=FLOAT(M)
  P=256.*X1+(X2+X3-2.)
  NM=8
  49 PP(NM)=P/8.**(NM-1)
  PP(NM)=AINT(PP(NM))
  P=P-PP(NM)*8.**(NM-1)
  PP(NM)=PP(NM)*10.**(NM-1)
  NM=NM-1
  IF(NM.LT.1) GO TO 52
  GO TO 49
  52 POCT=PP(1)+PP(2)+PP(3)+PP(4)+PP(5)+PP(6)+PP(7)+PP(8)
C PRINT DECIMAL WORD NUMBER OF CLOBBED WORD IN OCTAL
  WRITE(10,40) POCT
  40 FORMAT(" BAD KEY; WORD NUMBER=",F8.0," OCTAL")
C COUNT NUMBER OF ERRORS IN RECORD(N) AND IN KEY(N1)
  N=N+1
  N1=N1+1
C PRINT FAULTY KEYNAME OF CLOBBED KEY
  J1=M
  J2=M+4
  IF(N1.EQ.1) WRITE(10,58) (IBUF(NJ),NJ=J1,J2)
  58 FORMAT(" ",5A2)
C REPAIR FAULTY WORD
  IBUF(J+M-1)=NBUF(J+M-1)

```

Program Listing:

```
22 CONTINUE
C PRINT CORRECTED KEYNAME OF CLOBBED KEY
  IF(N1.EQ.0) GO TO 21
  WRITE(10,50) (IBUF(NL),NL=J1,J2)
50 FORMAT(" ",5A2)
21 CONTINUE
C REWRITE FAULTY DATAKEY0 RECORD
  IF(INDEX.EQ.1) GO TO 56
  IF(N.EQ.0) GO TO 56
  CALL WRB(ICHN,IBUF,K,4,IER9)
  CALL ERROR(IER9,"ERROR:WRB FOR DATAKEY0")
  WRITE(10,53) N
53 FORMAT(" RECORD REPLACED. NUMBER OF DISCREPANCIES IN RECORD=",
  *I7; /)
56 GO TO 20
C CLOSE CHANNELS TO DATAKEY0, BACKUP DISKETTE, AND SKEL
25 CALL KLOSE(ICHN,IER10)
  CALL ERROR(IER10,"ERROR:CLOSING DATAKEY0")
  CALL KLOSE(NCHN,IER11)
  CALL ERROR(IER11,"ERROR:CLOSING BACKUP DISKETTE")
  CALL KLOSE(ICHN12,IER15)
  CALL ERROR(IER15,"ERROR:CLOSING SKEL")
  WRITE(10,51)
51 FORMAT(" ", "PROGRAM FINISHED")
  STOP
  END
```

R

NOAA CENTRAL LIBRARY
CIRC QC874.3 U68 no.19
Tolleson, Pa DATAKEY repair program
3 8398 0003 3069 0

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

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 assess the socioeconomic impact of natural and technological changes in the environment and 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.

The major 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.

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

ATLAS — Presentation of analyzed data generally 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.

TECHNICAL SERVICE PUBLICATIONS — Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

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.



Information on availability of NOAA publications can be obtained from:

**ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE**

**6009 Executive Boulevard
Rockville, MD 20852**