

A
QC
874.3
U68
no.20

AA Western Region Computer Programs and
Problems NWS WRCP - No. 20

It



CONTIGUOUS FILE TRANSFER FROM THE DPCM TO THE DCM

Salt Lake City, Utah
September 1980

**U.S. DEPARTMENT OF
COMMERCE**

/ National Oceanic and
Atmospheric Administration

/ National Weather
Service



PREFACE

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 Transmissions of Raobs. John A. 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, Dec. 1979.
- 7 Plotting of Ocean Wave Energy Spectral Data. John R. Zimmerman, Dec. 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 Procedure. 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
- 19 DATAKEY Repair Program. Paul D. Tolleson, August 1980.

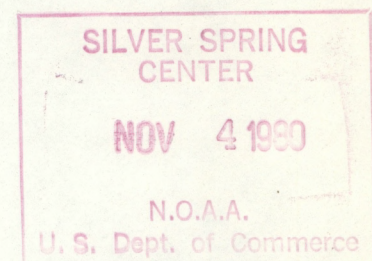
A
DL
874.3
268
no. 20

NOAA Western Region Computer Programs and Problems NWS WRCP - No. 20

CONTIGUOUS FILE TRANSFER FROM THE DPCM TO THE DCM

Paul D. Tolleson

National Weather Service Western Region Headquarters
Scientific Services Division
Salt Lake City, Utah
September 1980



UNITED STATES
DEPARTMENT OF COMMERCE
Philip M. Klutznick, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

National Weather
Service
Richard E. Halkgren, Director



CONTENTS

	<u>Page</u>
I. General Information	1
II. Application	1
III. Procedures	3

CONTIGUOUS FILE TRANSFER FROM THE DPCM TO THE DCM

Paul D. Tolleson*
Scientific Services Division
National Weather Service Western Region
Salt Lake City, Utah

I. General Information

A. Summary:

DPCMXFER.SV and DCMREC.SV are a package of utility programs that allow the user to transfer a contiguous file from the DPCM to the DCM through the MCA, provided the file exists on both sides. When the AFOS system is running in degraded mode on the DPCM side, the DPCM data files are updated while the DCM files become outdated. However, when the system is brought out of degraded mode and MODIFY is run, the DCM writes its outdated information over that of the DPCM. To overcome this problem, this package can be used to update LOCALDATA and DATAKEYØ on the DCM. AFOS may then be brought up with a minimum of data loss.

B. Environment:

DPCMXFER.FR, DCMREC.FR, and MURDER.FR have all been written in DATA GENERAL FORTRAN IV, for use specifically on the ECLIPSE used in AFOS. The program package is useful only at a WSFO where a DCM computer, a DPCM computer, and an MCA exist.

II. Application

A. Complete Program Description:

The purpose of this program package is to transfer a contiguous file from the DPCM to the DCM, provided the file exists on both sides. The program package consists of two save files, DCMREC.SV and DPCMXFER.SV, which must be invoked on the DCM side and DPCM side, respectively, and in that order.

DPCMXFER.SV is a multitasking program which contains two tasks; the main program called DPCMXFER.FR and a subordinate task called MURDER.FR. DPCMXFER.FR begins by reading a filename entered at the dasher and checking to see if the file is contiguous. If the file is not contiguous, the program terminates; if it is contiguous, the program sends the filename over the MCA to the DCM. After the filename has been received by the DCM, the main program opens the file in the DPCM and reads one block at a time, sending it over the MCA to the DCM. When an end of file is reached, zero bytes are sent over the MCA so that DCMREC.SV will record an end of file and close the MCA (see below). The main program closes all open channels and terminates.

*Mr. Tolleson worked as Summer Student Trainee in Scientific Services Division.

As the main program DPCMXFER begins to execute, it calls a task called MURDER. MURDER, given a higher priority than the main program, begins execution by opening a channel to the MCA (MCA receive) and waiting for an error return or message from DCMREC.SV. Meanwhile, as MURDER waits, the main program resumes execution as described above. Whenever an error return is received over the MCA from DCMREC.SV, MURDER is activated, terminating both the main program and the task. Messages will not cause termination. The CALL ERROR routine will terminate DCMREC.SV as well.

DCMREC.SV is a single task that interacts with DPCMXFER.SV. This program opens two channels to the MCA, one for transmitting and one for receiving. The transmission channel transmits error returns and program output to the MURDER task, while the receiving channel accepts data from the main routine in DPCMXFER.SV.

After opening the channels, the program reads the filename and checks to see if it is contiguous. If it is not contiguous, MURDER is activated and DCMREC.SV terminates; if it is contiguous, the program opens the outdated version of the file and reads via a loop the updated file from over the MCA in increments of one block, writing it over the outdated file. When zero bytes are transmitted over the MCA from DPCMXFER.SV, an end of file is recorded which allows the program to exit from the loop, close all channels, and terminate. Note that if any of the CALL routines in DCMREC.SV are unsuccessful, MURDER will be activated as described above.

Any special routines not found in the DATA GENERAL FORTRAN IV User's Manual have been taken from UTIL.LB (the utility library). They are listed below:

CALL ERROR	CALL OPENN	CALL RDS	CALL WRB
CALL GCHN	CALL WRS	CALL KLOSE	CALL RDB

B. Machine Requirements:

DPCMXFER.SV uses 11,264 bytes of storage, while DCMREC.SV uses 5,120 bytes of storage. Execution time depends on file size but can take several minutes for large files such as LOCALDATA.

C. Structure of Software:

The program package consists of two save files: DPCMXFER.SV and DCMREC.SV. DCMREC.SV is to be run on the DCM and DPCMXFER.SV is to be run on the DPCM. These two programs communicate with each other by passing files which exist on both sides from the DPCM to the DCM over the MCA.

Besides the main routine in DPCMXFER.SV, there exists an abortive task called MURDER. It waits to receive any error returns or messages over the MCA from DCMREC.SV. If it receives an error return, it terminates itself as well as the main routine.

There are no other subordinate tasks or subroutines.

D. Database:

The two programs operate only on the file specified during execution. It must exist both in the DCM and the DPCM and be contiguous. In addition, the file should be the same length on both sides. Two important files to which this applies are LOCALDATA and DATAKEYØ.

III. Procedures

A. Initiation of Program:

Initiate the program with the following procedure:

With the dasher switched to the DCM computer (P/K switch to B), go to the directory on which DCMREC resides.

DIR DPØ↵

Invoke the program.

DCMREC↵

Switch the dasher to the DPCM computer (P/K switch to A) and then go to the directory on which DPCMXFER resides.

DIR DPØ↵

Invoke the program.

DPCMXFER↵

B. Input Required:

After initiating the programs, DPCMXFER.SV will generate at the dasher the question,

FILENAME?

The user must enter the filename at the dasher.

The program will proceed without further input, accessing the file.

C. Output:

After DPCMXFER.SV and DCMREC.SV are initiated, and DPCMXFER.SV has read the filename, DCMREC.SV prints at the dasher

DCMREC:WAITING FOR FILENAME THROUGH MCA.

After it has received the filename, it prints

DCMREC:WAITING TO READ FILE THROUGH MCA.

After completing the transfer of the file, DPCMXFER.SV indicates its completion with

DPCMXFER:PROGRAM FINISHED.

Below is an example of dasher output and input. Note that the statement "FILENAME?" and "DCMREC:WAITING FOR FILENAME THROUGH MCA" were both generated by the programs before the filename was inputted. This occurs because of the higher priority given to MURDER.FR. Note that all underlined entries are user input.

```
DCMREC
DPCMXFER
FILENAME?
DCMREC:WAITING FOR FILENAME THROUGH MCA
DPØ:TEST
DCMREC:WAITING TO READ FILE THROUGH MCA
DPCMXFER:PROGRAM FINISHED
R
```

D. Cautions or Restrictions on Program Use:

This program must only be used on a contiguous file that is the same length both on the DCM and the DPCM sides. DCMREC.SV should be initiated first on the DCM, after which DPCMXFER.SV may be initiated on the DPCM. All messages outputted at the dasher labeled "DCMREC:" refer to DCMREC.SV and all messages labeled "DPCMXFER:" refer to DPCMXFER.SV.

E. Program Listing:

```

C INITIALIZE IBYT1,IBYT2
  CHANTASK 16.2
  EXTERNAL MURDER
  DIMENSION KIBUF(20),IBUF(256),INULL(2),ISTAT(18)
  CALL PRI(2)
  CALL ITASK(MURDER,1,0,IER0)
  CALL ERROR(IER0,"DPCMXFER:ERROR:ITASK FOR MURDER")
  IBYT1=40
  IBYT2=512
C READ FILENAME
  WRITE(10,100)
  100 FORMAT(" ", "FILENAME?")
  READ(11,101) KIBUF
  101 FORMAT(20A2)
C CHECK IF THE FILE IS CONTIGUOUS. IF NOT,
C TYPE A MESSAGE AND TERMINATE EXECUTION.
  CALL STAT(KIBUF,ISTAT,IER10)
  CALL ERROR(IER10,"DPCMXFER:ERROR:STAT FOR FILE")
  IF(ISTAT(7),3)) GO TO 99
  WRITE(10,104)
  104 FORMAT(" ", "DPCMXFER:YOUR FILE IS NOT CONTIGUOUS")
  WRITE(10,106)
  106 FORMAT(" DPCMXFER:PROGRAM ABORTED-KILL DCMREC WITH CONTROL A")
  GO TO 103
C OPEN CHANNEL TO MCA
  99 CALL GCHN(ICHN1,IER1)
  CALL ERROR(IER1,"DPCMXFER:ERROR:GCHN FOR MCA")
  CALL OPENN(ICHN1,"MCAT:2",0,IER2)
  CALL ERROR(IER2,"DPCMXFER:ERROR:OPENN FOR MCA")
C SEND FILENAME OVER MCA
  CALL WRS(ICHN1,KIBUF,IBYT1,IER3)
  CALL ERROR(IER3,"DPCMXFER:ERROR:WRS OF FILENAME TO MCA")
C OPEN CHANNEL TO FILE
  CALL GCHN(ICHN2,IER4)
  CALL ERROR(IER4,"DPCMXFER:ERROR:GCHN FOR FILE")
  CALL OPENN(ICHN2,KIBUF,0,IER5)
  CALL ERROR(IER5,"DPCMXFER:ERROR:OPENN FOR FILE")
C LOOP TO READ FILE AND SEND IT OVER MCA. IF AN
C END OF FILE IS ENCOUNTERED, GO TO STATEMENT 10 TO
C TERMINATE EXECUTION.
  NCNT=-1
  10  NCNT=NCNT+1
      CALL RDB(ICHN2,IBUF,NCNT,1,IER6)
      IF(IER6.EQ.9) GO TO 20
      CALL ERROR(IER6,"DPCMXFER:ERROR:RDS OF FILE")
      CALL WRS(ICHN1,IBUF,IBYT2,IER7)
      CALL ERROR(IER7,"DPCMXFER:ERROR:WRS OF FILE TO MCA")
      GO TO 10
C CLOSE MCA BY SENDING OVER 0 BYTES. CLOSE CHANNELS
C TO MCA AND FILE.
  20 CALL WRS(ICHN1,INULL,0,IER8)
  CALL ERROR(IER8,"DPCMXFER:ERROR:TERMINATING MCA")
  CALL KLOSE(ICHN1,IER9)
  CALL ERROR(IER9,"DPCMXFER:ERROR:KLOSE CHANNEL TO MCA")
  CALL KLOSE(ICHN2,IER10)
  CALL ERROR(IER10,"DPCMXFER:ERROR:KLOSE CHANNEL TO FILE")
  WRITE(10,102)
  102 FORMAT(" ", "DPCMXFER:PROGRAM FINISHED")
  103 STOP
  END

```


Program Listing (Continued)

```

TASK MURDER
DIMENSION NBUF(20)
C GET AND OPEN CHANNEL TO MCAR
  10 CALL GCHN(ICHN1,IER1)
  CALL ERROR(IER1,"SUBTASK ERROR:GCHN TO MCAR")
  CALL OPENN(ICHN1,"MCAR:2",0,IER2)
C WAIT, AND IF CHANNEL IS NOT AVAILABLE, GO
C TO STATEMENT 10 AND TRY AGAIN
  CALL WAIT(1,1,IER)
  IF(IER2.EQ.20) GO TO 10
  CALL ERROR(IER2,"SUBTASK ERROR:OPENN TO MCAR")
C WAIT FOR AN ERROR RETURN OVER THE MCA FROM
C DCMREC.SU. IF RECEIVED, THEN TERMINATE THIS
C TASK AND DPCMFER.SU
  30 CALL RDS(ICHN1,NBUF,40,IER3)
  IF(NBUF(1).NE."DC") GO TO 50
  WRITE(10,40) (NBUF(I),I=1,20)
  40 FORMAT(1X,20A2)
  GO TO 30
C PRINT OUT TERMINATION NOTICE AT THE DASHER
  50 WRITE(10,20)
  20 FORMAT(" DPCMFER:TERMINATED BY DCMREC.SU")
  STOP
  END

C INITIALIZE JBYT1,JBYT2
  DIMENSION JIBUF(20),JBUF(256),JSTAT(18),JERROR(20)
  JBYT1=40
  JBYT2=512
C OPEN CHANNEL TO MCAT
  CALL GCHN(ICHN3,IER11)
  CALL ERROR(IER11,"DCMREC:ERROR:GCHN FOR MCAT")
  CALL OPENN(ICHN3,"MCAT:1",0,IER12)
  CALL ERROR(IER12,"DCMREC:ERROR:OPENN FOR MCAT")
C OPEN CHANNEL TO MCAR
  CALL GCHN(ICHN1,IER1)
  IF(IER1.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:GCHN FOR MCAR
*    ",40,IER)
  JERROR(1)=IER1
  IF(IER1.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
  CALL ERROR(IER1,"DCMREC:ERROR:GCHN FOR MCAR")
  CALL OPENN(ICHN1,"MCAR:1",0,IER2)
  IF(IER2.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:OPENN FOR MCAR
*    ",40,IER)
  JERROR(1)=IER2
  IF(IER2.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
  CALL ERROR(IER2,"DCMREC:ERROR:OPENN FOR MCAR")
C TYPE A WAITING MESSAGE
  CALL WRS(ICHN3,"DCMREC:WAITING FOR FILENAME THROUGH MCA ",40,IER)
C READ FILENAME THROUGH MCA
  CALL RDS(ICHN1,JIBUF,JBYT1,IER3)
  IF(IER3.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:RDS FOR FILENAME OVER M
*CA ",IER)
  JERROR(1)=IER3
  IF(IER3.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
  CALL ERROR(IER3,"DCMREC:ERROR:RDS FOR FILENAME OVER MCAR")
C CHECK IF CONTIGUOUS FILE. IF NOT, TYPE
C MESSAGE AND TERMINATE EXECUTION.

```


Program Listing (Continued)

```

CALL STAT(JIBUF,JSTAT,IER10)
IF(IER10.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:STAT FOR FILE
*      ",40,IER)
JERROR(1)=IER10
IF(IER10.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
CALL ERROR(IER10,"DCMREC:ERROR:STAT FOR FILE")
IF(ITEST(JSTAT(7),3)) GO TO 99
CALL WRS(ICHN3,"DCMREC:YOUR FILE IS NOT CONTIGUOUS      ",40,IER)
CALL WRS(ICHN3,"DCMREC:PROGRAM ABORTED                  ",40,IER)
CALL WRS(ICHN3,JERROR,40,IER)
GO TO 103
C OPEN CHANNEL TO FILE
99 CALL GCHN(ICHN2,IER4)
IF(IER4.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:GCHN FOR FILE
*      ",40,IER)
JERROR(1)=IER4
IF(IER4.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
CALL ERROR(IER4,"DCMREC:ERROR:GCHN FOR FILE")
CALL OPENN(ICHN2,JIBUF,0,IER5)
IF(IER5.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:OPENN FOR FILE
*      ",40,IER)
JERROR(1)=IER5
IF(IER5.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
CALL ERROR(IER5,"DCMREC:ERROR:OPENN FOR FILE")
C TYPE A WAITING MESSAGE
CALL WRS(ICHN3,"DCMREC:WAITING TO READ FILE THROUGH MCA ",40,IER)
C LOOP TO READ OVER MCA. EXIT WHEN AN END OF FILE IS REACHED
NCNT=-1
30 CALL RDS(ICHN1,JBUF,JBYT2,IER6)
IF(IER6.EQ.9) GO TO 40
IF(IER6.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:RDS FOR MCA
*      ",40,IER)
JERROR(1)=IER6
IF(IER6.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
CALL ERROR(IER6,"DCMREC:ERROR:RDS FOR MCA")
NCNT=NCNT+1
CALL WRB(ICHN2,JBUF,NCNT,1,IER7)
IF(IER7.NE.1) CALL WRS(ICHN3,"DCMREC:ERROR:WRB FOR FILE
*      ",40,IER)
JERROR(1)=IER7
IF(IER7.NE.1) CALL WRS(ICHN3,JERROR,40,IER)
CALL ERROR(IER7,"DCMREC:ERROR:WRB FOR FILE")
GO TO 30
C CLOSE CHANNELS TO MCA AND FILE
40 CALL RESET
103 STOP
END

```


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**