NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

Special Purpose
Utility Programs

N.I.O. Internal Report No. N. 29

1973

NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY.

SPECIAL PURPOSE

UTILITY PROGRAMS

N.I.O. Internal Report No. N29.

1973.

N.I.O. PROGRAMS N.29

(Subroutines have negative numbers)

Disk and Magnetic Tape Utilities

-51	Search FLET	FLETM
-155	To read or write one disk sector from Process Work Storage	DRED
-192	Allow a write to a file protected disk	WRIFT
-193	Disallow a write to a file protected disk	WRAFT
-204	To reverse a disk file for use with R.J.E.	REVRS
-205	To generate BCD magnetic tapes in a Fortran program	PAPEB
193	Magnetic tape copying utility program	TAPCO
274	To convert disk files to R.J.E. format	CONDS

Special Utility Routines

-189	Check for invalid day, hour, minute	ITCHK
-191	Convert time in 1/10th minute to hours and minutes	LTIME
263	Telephone directory	PHONE

File Manipulation

207	Correction of times in ship's data files	COFIT
208	Correction of wind and speed in ship's data file	COMET
209	Apply depth corrections to ship's data files	CODFI
245	Analyse, plot and print meteorological data	PAM
246	Edit ship's data files conversationally	SYSED
266	Correct depth in ship's data files	COD
270	Update ship's depth files	UPDEP
277	Check for time gaps in depth files	DATCK

Queries regarding the use or availability of
any of the programs in this volume may be made to:-

The Program Librarian,
Data Processing Group,
National Institute of Oceanography,
Wormley, Godalming, Surrey.

from whom a comprehensive list of all current N.I.O.
Programs is available.

All the programs in this volume have been compiled and executed on an I.B.M. 1800 Computer having the following configuration:-

1802 Processor-controller with 32,768 words of core storage

3 1810 Disk Drives Model A

2 2401 Magnetic Tape Drives (30 Kc/s) (7 track)

1443 Printer, 240 lines/minute

1442 Model 6 Card Read-Punch

1816 Keyboard-Printer

Facit Paper Tape Reader, 1000 Characters/second

Facit Paper Tape Punch, 150 Characters/second

The Operating Systems were T.S.X. Version 3 or M.P.X. Version 3.

N.I.O. Program -51

Program Title: (For Lib. List)	SEARCH FLET
Classification Letter:	S
Program Name:	FLETM
Version Number:	V21
Type of Program:	Subroutine
Operating System:	MPX
Language:	Assembler
Programmer/date:	T. Voss 21.8.72
Program Passed By:	R. Wells 13.10.72
Description:	Searches FLET on specified drvie for a named file and returns address size and type of core load.
System Prerequisites:	MAIN PROGRAM MUST HAVE *IOCS(DISK)
Inskel Common Variables:	
Subroutines Called:	GTBUF, FREUP, BULKN
Programs Called:	
Files Called:	
Operation and Method:	

USE: CALL FLETM(NAME,IDRIV,ISEC,IWD,ITYPE)

ON CALL: NAME is name of required file (3A2)

            IDRIV is drive to be searched

ON RETURN: ISEC is drive code and sector address (DSA) of requested core-load, or Ø
            if not found

            IWD is word count of core-load or sector count of data file or Ø if not
            found

            ITYPE is type of file located

            Ø   data file

            1   batch core-load

            2   process core-load

            3   spar core-load

Method   This version of FLETM is based on the version 2Ø program by J. Crease.
            Signigicant changes have been made to the program logic as follows:-

1. The program selects as priority for disk operations its execution level (but 15).
   Thus if FLETM is called from a program running on level 8, it uses disk priority 8.

2. This version of FLETM uses a fortran disk buffer as work space.
   The main program must have a *IOCS(DISK) statement. This saves about 310 words in
   a program that would have had this card in any case.

N.I.O. PROGRAM -155

Program Title: (For Lib. List)	To read or write one disk sector from process work storage
Classification Letter:	S
Program Name:	DRED and DRIT
Version Number:	1
Type of Program:	Either Process or Non-process
Operating System:	TSX
Language:	1800 Assembler
Programmer/date	B.D.Page
Program Passed By:	B.D.P.
Description:	Identical to NIO -49 (DREAD and DRITE) except that the busy tests have been removed after the disk operations in order not to hold up system.
Inskel Common Variables:	None
Subroutines Called:	DISKN
Programs Called:	None
Files Called:	None
Operation and Method:	The subroutine is called as follows:-

CALL DRED(IDISK,ISECT,IFILE) or

CALL DRIT(IDISK,ISECT,IFILE) where

IDISK is the drive number (=0, 1 or 2)

ISECT is the sector to be read or written in process work storage.

IFILE is the array for data transfer and must be dimensioned IFILE(322), the last two elements are used by DRED.

N.I.O. PROGRAM −192

Program Title: (For Lib. List)	Allow file-protected write to disk
Classification Letter:	S
Program Name:	WRIFT
Version Number:	20
Type of Program:	Nonprocess
Operating System:	MPX or MPXPT
Language:	1800 Assembler
Programmer/date:	W. Strudwick/26.4.72
Program Passed By:	R. Wells
Description:	To allow BULKN to write to file-protected areas of disk
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	
Programs Called:	
Files Called:	
Operation and Method:	

Word 35 of the level work area is set to non-zero.

The subprogram is called by the statement

CALL WRIFT

Note Subprogram WRAFT should be called after writing file-protected data.

N.B. D.P.G. internal use only

N.I.O. PROGRAM -193

Program Title: (For Lib. List)	Dis-allow file-protected write to disk
Classification Letter:	S
Program Name:	WRAFT
Version Number:	20
Type of Program:	Non-process
Operating System:	MPX or MPXPT
Language:	1800 Assembler
Programmer/date:	W. Strudwick/26.4.72
Program Passed By:	R. Wells
Description:	To prevent BULKN writing file-protected data to disk

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called:

Operation and Method:

Word 35 of the level work area is set to zero.

The subprogram is called by the statement

CALL WRAFT

<u>Note</u> This nullifies the effect of subprogram WRIFT.

<u>N.B.</u> D.P.G. internal use only

N.I.O. PROGRAM -204

Program Title: To reverse a file on disk for use with disk RJE
(For Lib. List)

Classification Letter: S

Program Name: REVRS

Version Number: 20

Type of Program: Subroutine

Operating System: MPX

Language: Fortran IV

Programmer/date: M. Olliff 2.10.72

Program Passed By: E. Page

Description: Reads a sector from disk, reverses it and writes it back

System Prerequisites:

Inskel Common Variables:

Subroutines Called: RVERS

Programs Called:

Files Called:

Operation and Method:

To use, CALL REVRS(LREC,IOUT)

where IOUT is an array of 320 words and LREC contains the number of the last record

plus one of the disk file which needs to be reversed. A define file statement of the

form 2(N,40,U,IREC) must be set in the calling program. N is the number of records

in the file. The logical file number must be 2. As the subroutine is for use when

setting up files for RJE disk work, the records are assumed to be 40 words long.

A sector at a time is read from disk, reversed and written back. This is repeated until

LREC records have been reversed.

N.I.O. PROGRAM -205

Program Title: (For Lib. List)	BCD Magnetic Tape Subroutines
Classification Letter:	S
Program Name:	PAPEB, BDSET, BDEOF, CLOSE, PAPTN
Version Number:	20
Type of Program:	Nonprocess
Operating System:	MPX but not MPXPT
Language:	1800 ASSEMBLER
Programmer/date:	W.K. Strudwick / 7.3.73
Program Passed By:	
Description:	S/R's to allow the user to read and write BCD
System Prerequisites:	magnetic tapes using FORTRAN formatted I/O. IBM supplied PAPTN and PAPEB must not be in the executive.
Inskel Common Variables:	None
Subroutines Called:	BCDA1, A1BCD, ZZIPP, MAGT
Programs Called:	
Files Called:	

Operation and Method:

These subroutines replace the IBM supplied PAPTN and PAPEB and allow the user to read or write BCD magnetic tapes by referencing logical unit number 4.   Three special subroutines are used by the programmer they are used thus:-

CALL BDSET (MODE, IREC, NOREC)

where MODE is set up in a DATA statement as follows:-

DATA MODE/ZØABC/
where

$$A=\emptyset \text{ for odd parity}$$
$$=1 \text{ for even parity}$$
$$B=4 \text{ for 800 bpi}$$
$$=5 \text{ for 200 bpi}$$
$$=6 \text{ for 556 bpi}$$
$$C=\emptyset \text{ for first tape drive}$$
$$=1 \text{ for second tape drive}$$

IRECL = is the record length in characters
NOREC   is the number of records per block. This subroutine must be called before writing commences.

CALL BDEOF writes an E-O-F marker
CALL CLOSE writes two E-O-F markers and rewinds the tape.

An *IOCS (PAPER TAPE) record must be included in the deck.

N.I.O. PROGRAM 193

Program Title: (For Lib. List)	Tape copying utility program.
Classification Letter:	S
Program Name:	TAPCO
Version Number:	2∅
Type of Program:	Non-Process
Operating System:	M.P.X.
Language:	18∅∅ Fortran IV
Programmer/Date:	Eileen Page/January 1972
Program Passed By:	Eileen Page
Description:	To copy magnetic tape files or complete tapes from tape drive 1 to tape drive 2.
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	DATSW, FILE2, MAGOP, STORM, UNPAC, NCOMP
Programs Called:	
Files Called:	MFILE

Operation and Method:

Input:
```
        //∅JOB
        //∅*PROJECT/NAME/TITLE
        //∅XEQ∅TAPCO∅∅∅FX
```

followed by 1 parameter card for each set of consecutive files to be copied from tape A to tape B, in the following format:-

```
cc1     4      9    12       17   20   23   26   29   32
   X b b N A M E T 1 b b N A M E T 2 b b M O D 1 b b M O D 2 b b M b b
cc35      39   42     46   49     53   56   59
   I F R S T b b I L A S T b b J F R S T b b B L C K
```

where:-
a) X is set to * for the last set of files to be copied in the job.
b) NAMET1 is the name, left justified, of the tape to be copied from.
c) NAMET2 is the name, left justified, of the tape to be copied to.
d) MOD1 is the decimal mode of the tape to be copied from.
e) MOD2 is the decimal mode of the tape to be copied to.
   MOD1 and MOD2 are set up as follows (starting with the left-hand digit):-
   Digit 1 = ∅
   Digit 2 = ∅ for odd parity tape
           = 1 for even parity tape

Digit 3 = ∅ for 8∅∅ b.p.i.)

= 1 for 2∅∅ b.p.i.) 3 characters/word

= 2 for 556 b.p.i.)

= 4 for 8∅∅ b.p.i.)

= 5 for 2∅∅ b.p.i.) 2 characters/word

= 6 for 556 b.p.i.)

Digit 4 = ∅ for tape drive 1 (LUN 5)

= 1 for tape drive 2 (LUN 1∅)

f) M is set to M if the tape to be copied to operates under the N.I.O.
Tape security system, otherwise blank.

g) IFRST is the number of the first file to be copied from tape A.

h) ILAST is the number of the last file to be copied from tape A. ($\geq$ IFRST)

i) JFRST is the number of the first file to be copied onto tape B.

j) BLCK is the tape block length in words. e.g. 4∅∅∅ char/block at
2 char/word gives a block length of 2∅∅∅ words.
In 18∅∅ Fortran on 7 track tape, a MODE of ∅∅∅∅ is used, giving
3 bytes/word. The total number of characters written per record is 146
and a one word header, therefore the block length is 146/2+1 = 74 words.
(There are 2 characters per word, but in formatted Fortran on magnetic
tape, each word is split up into 3 bytes).

Output:- The required files are written to the tape on tape drive 2. If tape B
operates under T.A.S.S. the appropriate file or disk is automatically
updated.

Method:- The two modes are converted to hexidecimal by multiplying each element by
the appropriate power of 16, and the (BLCK+1) the element of the array
LIST is set to BLCK, and all preceding elements are set to zero.
A message is printed to the typewriter asking the operators to load the
correct tapes to the tape units. The system pauses while this operation
is carried out.
The program then tests for M in column 32 of the parameter card. If it
is present the MFILE file on disk, which contains all the entries for the
magnetic tapes operating under T.A.S.S., is searched to find the entry for
tape NAMET2. If it is not found (∗) occurs and the program is aborted.
When the entry is found, the next available file for that tape is tested
against JFST, and if some tape is about to be overwritten, a message to
the effect is written to the typewriter. If overwriting is intended, data
switch 9 must be turned on and the system started, otherwise the job will
be aborted.
Both tapes, whether they operate under T.A.S.S. or not, are positioned at
files IFST and JFST, and the number of files to be copied (ILST-JFST+1)
is calculated.
∗ the error "MDFIO BAD REC NO"

Copying then commences, block by block, from drive 1 to drive 2. After each
block has been read, the possible errors set up in MAGOP are tested. When
an incorrectable tape error is found, a message to this effect is printed on
the 1443 printer and the two tapes are rewound. This error routine also
occurs if the wrong length block has been requested for the tape on drive 1.
If an error was detected and corrected while reading or writing a message is
again printed out and the program continues. When a correct block has been
read, the block number is increased by 1 and the block is written to drive 2.
The tapes are again rewound if an uncorrectable error occurs, or if the end
of tape is detected.

When an end of file marker is encountered on drive 1 an endfile marker is
written to drive 2, and if the tape on drive 2 operates under T.A.S.S. the
the disk file record for that tape is updated.

Further sequential files are then copied in a similar manner, and when all
the files specified in the parameter card have been copied the two tapes
are rewound and the next parameter card is read in. When the card containing
* in column 1 has been executed, the program exits.

Notes: If Tape 2 operates under T.A.S.S. but 'M' is not present in the parameter
          card, the tapes will be copied but the disk file will remain unaltered.

Operator Instructions: If overwriting of tapes is permissable set data switch 9 on
                        and press start when the system executes PAUSE 2222.
                        At a PAUSE 1111, load the tapes requested in the typewriter
                        message. In all cases reset all data switches as soon as the
                        system has restarted.

N.I.O. Program 274

Program Title: (For Lib. List)	To convert data files on disk to be run on the RJE link
Classification Letter:	S
Program Name:	CONDS
Version Number:	2Ø
Type of Program:	PROGRAM
Operating System:	M.P.X.
Language:	Fortran IV
Programmer/date:	M. Olliff 23.5.72
Program Passed By:	R. Wells
Description:	Conversion of data files on disk to character form
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	DFT, DFTCH, CONF, IPUT, CONE, PACC
Programs Called:	
Files Called:	
Operation and Method:	

The program reads data from a disk file, converts it into card image format and
stores it in another file for use on the 360 via RJE18.

The data can be in any form, as the program uses subroutines CONE, CONF, IPUT to
convert the data in to packed character form.

The characters are packed into 40 word records. A printout of the data after
conversion can be requested. The last record in the file will contain '/END' in
A2 for use by RJE18.

N.B. The output records will be larger in most cases after conversion, for example
a six digit integer on disk normally occupying one word, would be 6 characters long,
occupying three words.

All input files should be on the same disk, if they are to be used in one run of the
program.

Inputs from cards

```
             cc    14      19    24
//ØJOB           X   16   X     X
//ØXEQ CONDS         FX
```

followed by data cards in the form

1st Card   cc1   7

          XXXXXXYYYYYY

where XXXXX is the no. of files to be converted, decimal, right justified
and YYYYYY is the print indicator. If a listing of the data is required, this
should be set to one, if not required, leave blank.

This card is followed by sets of cards, one set for each file to be converted, made up of -

1 card in the form

```
cc1     7    13    19    25    31    37
   AAAAAABBBBBBXXXXXXYYYYYYCCCCCCZZZZZZDDDDD
```

where AAAAAA is the name, left justified of the data file to be converted

BBBBBB is the name, left justified of the data file to receive the data

XXXXXX is the no. of records to be converted, right justified, of the input
file

YYYYYY is the no. of words per record, right justified, of the input file

CCCCCC is the drive no. of the input file decimal, right justified

ZZZZZZ is the no. of records on the output file, decimal, right justified

DDDDDD is the drive no. of the output file, decimal, right justified

and as many cards as are required in the form:-

```
cc1 6    13    20    27    34    41    48    55    62    69
XXXXXAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.YAAPYY.Y
```

where XXXXX is the no. of records of this format (on this card) to be converted
(consecutively)

and

AAPYY.Y is the format type, AA is the no. of this type, P is the 'type', and
YY.Y is the size.

10 different formats are allowed per record.

Output

Output is a data file in Card Image Format, and a printout of the data if required.

Timing

The program takes approximately 30mins. to convert 3750 Output records without a
listing.

Example

A file called FIL1 contains 5Ø records, the first of which is a header record
containing a 7Ø character comment. The next 5 records contain 1Ø integer numbers
per record, and the remaining contain 8 real numbers per record. A printout is
required. Record size on input is 35 words. The output file is called OUTF1 and
the records are always 4Ø words long (card image). FIL1 is on drive 2 disk, OUTF1
is on drive 1. The length of OUTF1 may be calculated as follows:-

The header will need 2 output records as 4Ø < 7Ø < 8Ø

one record 2 records

Allowing 6 digits per integer, the next 5 input records will each contain 6 x 1Ø
characters per record, needing 2 output records each. The remaining 44 records,
allowing 1Ø digits per real no. will each contain 8 x 1Ø chars. per record, needing
2 output records each. 1 record is then required for the /END statement, giving
101 records in all.

There are 8 records to a sector on output, so OUTF1 should be set up with 13 sectors. The data cards required are as follows:-

```
          cc1    6    12
```

Card 1   ﾟﾟﾟﾟ1ﾟﾟﾟﾟ1   for 1 file with listing 42

Card 2   FIL1ﾟﾟOUTF1ﾟﾟﾟﾟﾟ5ﾟﾟﾟﾟﾟ35ﾟﾟﾟﾟ2ﾟﾟﾟ1ﾟ1ﾟﾟﾟﾟﾟ1

Card 3   ﾟﾟﾟﾟ135Aﾟ2.Ø   (header record)

Card 4   ﾟﾟﾟﾟ51ﾟIﾟ6.Ø   (integer records)

Card 5   ﾟﾟﾟ44ﾟ8Fﾟ10.3   (real no. records)

N.I.O. PROGRAM -189

Program Title: Check for invalid time (DAY,HOUR,MIN)
(For Lib. List)

Classification Letter: S

Program Name: ITCHK

Version Number: 20

Type of Program: Function Subroutine

Operating System: TSX/MPX

Language: Fortran IV

Programmer/date: J. Sherwood Feb 1972

Program Passed By: E. Page

Description:

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called:

Operation and Method:

The function

                                ITCHK(IDAY,IHOUR,IMIN)

checks that   IDAY lies between 1 and 366

                                  IHOUR lies between 0 and 23

                                  IMIN lies between 0 and 59

and returns   ITCHK = 0 if valid

                                      = 1 if invalid

****

This write-up replaces TCHK (-189)

N.I.O. PROGRAM -191

Program Title: (For Lib. List)	Convert time 1/10 min. to hours/mins for output
Classification Letter:	S
Program Name:	LTIME
Version Number:	20
Type of Program:	Subroutine
Operating System:	TSX/MPX
Language:	Fortran IV
Programmer/date:	J. Sherwood Feb 72
Program Passed By:	E. Page

Description:

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called:

Operation and Method:

The call

CALL LTIME(L,IHR1,IHR2, IMIN1, IMIN2)

where L is time in 1/100 min.   (L must lie between 0 and 14399)

returns four integers

IHR1   : tens of hours

IHR2   : hours

IMIN1 : tens of minutes

IMIN2 : minutes

These may now be output in 4I1 format so that leading zeros are retained.

The returned time is rounded to the nearest minute. L is unchanged.

N.I.O PROGRAM 263

Program Title: (For Lib. List)	N.I.O. Telephone Directory
Classification Letter:	S
Program Name:	PHONE
Version Number:	2∅
Type of Program:	Non-Process
Operating System:	M.P.X.
Language:	FORTRAN IV
Programmer/Date:	P. Lindsay/1.5.72
Program Passed By:	C. Spackman
Description:	Prints out the N.I.O. Telephone Directory
System Prerequisites:	None
Inskel Common Variables:	None
Subroutines Called:	None
Programs Called:	None
Files Called:	None

Operation and Method: The program reads in from cards all information concerning the telephone extension numbers, the rooms and personnel. From this the user can specify one of the three forms to present this data.

1) Any pages concerning special rooms followed by an alphabetic list of the personnel. For this latter part an index character is provided on the right of the page to allow the eventual user quick access to the respective pages.

2) A combined list (rooms and personnel given in room number order).

3) A combined list given in telephone number order.

Use:

```
        //∅JOB
        //∅FOR∅PHONE

                SOURCE PROGRAM

        //∅XEQ∅PHONE
        *CCEND

                DATA Part 1(presorted alphabetically)
    999
                DATA Part 2(presorted alphabetically)
    999
```

where Part 1 consists of the 'Special Rooms Data', which gives the information about significant rooms or departments within the Oceanography.

Part 2 consists of the 'Personnel Data', giving the information about the whereabouts of the people within the Oceanography.

Within the data (Part 1) a blank card indicates a throw to a new page. The number 999 (in card column 1-3) acts as the terminator for both sets of data.

Format of Data:-

	Columns	1 - 3	Room Number
		4	Alphabetic suffix to the room number
		8 - 33	Name of the person or room
		35 - 37	Telephone extension number (if any)

N.B. An extra card is required on the end of the data deck because of the use of CBUF (card buffering program).

N.I.O. PROGRAM 207

Program Title: (For Lib. List)	Correction of times on the files DATF and GEOF
Classification Letter:	S
Program Name:	COFIT
Version Number:	21
Type of Program:	Non-Process
Operating System:	M.P.X.
Language:	18ØØ FORTRAN IV
Programmer/Date:	W. Strudwick/10.4.72
Program Passed By:	C. Spackman
Description:	COFIT scans through files DATF and GEOF and checks that the times in each correspond one to one
System Prerequisites:	None
Inskel Common Variables:	None
Subroutines Called:	JFND, DFT
Programs Called:	None
Files Called:	User option

Operation and Method:

The program is stored on a drive 1 disk and may be invoked by the cards:-

```
cc1          14    19    24
//ᵇJOB        X     X     X
//ᵇXEQᵇCOFITᵇᵇᵇFX
```

Input:

The first card contains the names of the files in the format:-

cc

1 - 5   Name of DATF file, left justified.

7 -11   Name of GEOF file, left justified.

The second card contains the record number, on the GEOF file, at which the check of times is to begin in columns 1 to 4 integer right justified. If the whole of the file is to be searched this value should be zero.

Output:

There are four possible cases.

1) If the times on both files exactly correspond and are all increasing with record number there is no output.

2) If two successive times on the GEOF file are not in ascending order the following message is printed

TIMES NOT IN ASCENDING ORDER AT RECORD n. THIS RECORD HAS BEEN NEGATED

The record number indicated will be the second of the records out of sequence.

3) If a time on DATF is different from on GEOF but the times in the following
records on both files are the same then the following message will be printed.
TIMES ON GEOF AND DATF DIFFERENT THE DAY NUMBERS HAVE BEEN NEGATED
By this is meant the day numbers of the erroneous records on files DATF and
GEOF have been made negative.

4) If, however, a time on GEOF is different from a time on DATF and also the times
in the following records are also different then the program assumes that a record
is missing on either GEOF or DATF. The program checks to see which case applies and
one of the following messages will be printed.
a) RECORD TOO MANY FILE GEOF. LAST GEOF RECORD IN STEP = N
b) RECORD TOO MANY ON FILE DATF. LAST DATE RECORD IN STEP = N
In both cases this will be followed by the message
DATF TIMES AFTER THIS RECORD ARE xxxx   xxxx
GEOF TIMES AFTER THIS RECORD ARE xxxx   xxxx
Where xxxx are the values of the times mentioned in 1/10 minute. In this last
case the program will exit without checking the rest of the file. Program IDODI
(N.I.O. 213) should then be used, using the information given in message (a) or
(b) to bring the GEOF and DATF files back into step. COFIT should then be run
again to check the rest of the files.

Execution Time Approximately five minutes per file.

N.I.O. PROGRAM 208

Program Title:-	Correct GEOF files
(For Lib. List)	
Classification Letter:-	S
Program Name:-	COMET
Version Number:-	21
Type of Program:-	Non-Process
Operating System:-	M.P.X.
Language:-	18ØØ FORTRAN IV
Programmer/Date:-	W. Strudwick / 10.4.72
Program Passed By:-	C. Spackman
Description:-	COMET is used for loading the corrected wind speed, wind direction and magnetic anomaly onto GEOF using the raw data from DATF
System Prerequisites:-	None
Inskel Common Variables:--	None
Subroutines Called:-	MAGFD,RGMAG,JFND,FXDEG,DFT
Programs Called:-	None
Files Called:-	User option

Operation and Method:- The Program is stored on a drive 1 disk and is invoked by the cards:-

```
        cc1              14   19   24
          //ƒJOB         X    X    X
          //ƒXEQƒCOMETƒƒƒFX
```

Input:-   First card contains the names of the files in the format:-

    cc

        1-5 Name of DATF file, left justified

        7-11 Name of GEOF file, left justified

    Second card:-

    cc

        1-4 Day number of first record to be processed, integer right
             justified.

        5-10 Time of first record to be processed, integer right justified.

        11-14 Day number of last record to be processed, integer right
             justified.

        15-20 Time of last record to be processed, integer right justified.

The times must be in 1/1Ø minute and correspond to times on the GEOF file.

        Third card:-

            cc1 = zero if magnetic anomaly not required.

                non-zero if magnetic anomaly required.

Fourth card:-

        cc

        1-4 YEAR (e.g. 1972)

Output:-   The first line of output gives the two times between which the
corrected windspeeds and magnetic anomalies have been calculated.
Then follows a table containing the information which has been
written to GEOF:-
Day, time windspeed, wind direction and magnetic anomaly. If input
times cannot be found on the files, a message to this effect is
printed out.

Method:-   The method used is similar to that for METC.

Execution Time:-   Approx. 6 + N minutes where N = No. of hours of data.

N.I.O. PROGRAM 209

Program Title:-	Depth Corrections to GEOF file
(For Lib. List)	
Classification Letter:-	S
Program Name:-	CODFI
Version Number:-	21
Type of Program:-	Non-Process
Operating System:-	M.P.X.
Language:-	1800 FORTRAN IV
Programmer/Date:-	N. Olliff and W. Strudwick
Program Passed By:-	C. Spackman
Description:-	CODFI is an adaption of DFIL for running off-line and is used for copying information from a corrected depth file onto the relevant positions on GEOF.
System Prerequisites:-	None
Inskel Common Variables:-	None
Subroutines Called:-	JFREC, JEND, DFT
Programs Called:-	None
Files Called:-	User option

Operation and Method:-

The program is stored on a drive 1 disk and is invoked by the cards:-

```
cc1            14    19    24
  //ЬJOB        X     X     X
  //ЬXЕQЬCODFIЬЬЬFX
```

Input:-

The first card contains the names of the data files to be used in the following format:-

cc

1 - 5 Name of GEOF file, left justified

7 -11 Name of file containing the corrected depths in metres, left justified

The second card contains the following:-

cc

1 - 4 Day number of first record to be processed, integer right justified.

5 -10 Time of first record to be processed, in tenths of minutes, integer right justified.

11-14 Day number of last record to be processed, integer right justified.

15-20 Time of last record to be processed, in tenths of minutes, integer right justified.

OUTPUT:-   The first line of output gives the two times between which the
                   depth readings have been called. Then follows a table containing
                   the information written to GEOF which is laid out as:-
                   Time on depth file, Time on GEOF, corrected depth.

                   Error messages are output for the cases where the data cannot be
                   found on either of the files, and also where the finishing time
                   is after the end of either file.

METHOD:-   The depth file is searched through until a time greater than or
                   equal to the starting time is found. Then the relevant depth is
                   written into the nearest time on GEOF. If a GEOF record cannot
                   be found that is within 2 minutes of the time of the depth entry
                   the program will ignore that depth and read the next depth entry
                   in the file.

EXECUTION TIME:- Approximately five minutes for 24 hrs. data.

Title   Analyse, Plot and Print Meteorological data from

                               ships data files.

Name   PAM

Machine   IBM 18ØØ

Type of Program   Non-process

Language   18ØØ FORTRAN IV

Operating System   T.S.X. or M.P.X.

Subroutines Called   WNREL, DFTCH, DFT

Purpose   To calculate the meteorological data collected on

                               board ship which is stored on data files and to

                               print and/or plot the derived data.

Use   To use PAM punch a set of cards as follows (assuming

                               program is stored on disk):

                                                        19   24

                               // b JOB n   x   x
                               // b * PROJECT NUMBER/NAME/TITLE
                               // b XEQ b PAM
                               * RILES (1,, n) (2,, n)
                               * CCEND
                               Where n is the drive number on which the tempory

                               storage disk resides.

                               The input parameter cards follow these cards as

                               described under input.

Input (i) Calibration Factors card

cc 1 - 7 Starboard Dry-Wet multiplicative factor

8 - 14 Port Dry- Wet multiplicative factor

15 - 21 Port Dry multiplicative factor

22 - 28 Port Dry additive factor

29 - 35 Starboard Dry multiplicative factor

36 - 42 Starboard Dry additive factor

Each factor is in the form XXX.XXX

(ii) File specifications card.

cc 1 - 5 Name of file, left justified.

7 Drive Number

9 - 12 First record number, right justified

14 - 17 Last record number, right justified

19 - 2∅ Time increment in minutes, right justified.

22 = 1 if another card is to follow else blank
or zero.

(iii) More of type (ii) if more than one data file is to be
processed in one run.

(iv) Option card.

cc 1 = ∅ or blank for Plot & Print

= 1 for Print only

= 2 for special output option

cc 3 - 4 Used only if cc1 = 2

= ∅2 for output on cards

= ∅4 for output on paper tape

= ∅5 for output to magnetic tape on deck 1.

= 1∅ for output to magnetic tape on deck 2.

(v) Plot parameter card (only used if option card contains
∅ or blank)

cc 1 - 2 Minimum temperature, right justified

7   Distance in inches between minimum and

maximum temperatures.

9 - 13, 14 - 18, 19 - 23 .... 44 - 48

Distances between end of previous graph and

beginning of next in inches.

Each value is of the form xx.xx A positive

value will cause overlapping of the graphs.

Note:

cc 44 - 48 Contain the distance from the last graph

to the start of the day numbers annotation.

Output   The first page of the output gives the calibration

factors being used in the program and the day numbers

and times of the first and last records in the file

being processed.

The remaining output is dependant on which option has

been spectfied.

Error Messages   Messages will be output when the file being called

for is either not on the disk or the header of the

file is wrong.

Programmer   WILLIAM STRUDWICK

Title: Edit ship's data files conversationally.

Name: SYSED

Language: 18∅∅ Fortran IV

Machine: I.B.M. 18∅∅

Operating System: M.P.X. , T.S.X. , M.P.X.P.T. , T.S.X.P.T.

Purpose: To edit ship's data files.

Input: The input to SYSED is via the keyboard of the typewriter.
                      The word READY is typed by the program and then the user
                      may enter any one of the following commands.
                      (Note δ is the EOF Key. Characters underlined are
                      optional.)
                      1) FILE$\emptyset$XXXXXδ
                         where XXXX is the name of the file to be edited.
                         e.g. To gain access to file DATF1 type
                         FILE$\emptyset$DATF1δ
                         This command causes the in-core define file table for
                         SYSED to be dynamically altered. The header of the
                         file is read into core and checked. If the header is
                         wrong the following message is typed:-
                                FILE OR RECORD LENGTH WRONG, CHECK HEADER
                         This error prohibits the use of this file in the program
                         until the header is changed using the HEADER command.
                      2) HEADERδ
                         This causes the header of the file set up using the FILE
                         option to be typed out.
                         After the header has been output the following message
                         appears:-
                                DO YOU WISH TO CHANGE ANY VALUES?
                         The user may then type either YESδ or NOδ
                         If the response is YESδ then the program replies with
                                ENTER NAME OF VALUE YOU WISH TO CHANGE
                         The user may then enter
                                NTOT δ, NEXTδ , NAME δ, NWORD δ, or ICR δ.
                         The program then types:-
                                ENTER VALUE OR ENTER NAME
                         (depending on what is required)
                         The user then enters the new value in free format, left
                         justified, followed immediately by the E-O-F character.
                         The program then responds with
                                ANY MORE CHANGES
                         The user may type YES δ or NO δ and the program responds
                         accordingly.
                         If, for this file, a wrong header message has been
                         typed the following message is output:-
                                FILE RE-DEFINITION IN PROGRESS
                         and the program tries to set the file up again. If it
                         is successful the file may be used.

3) TYPEⱠrec1-rec2,rec3-rec4,.....recn-recn ⌀
where rec1,rec2,rec3,rec4....recn,recn are record
numbers.
This command causes the records between rec1 and rec2,
rec3 and rec4,....and recn and recn to be typed out.
The first number on each line is the record number.
e.g. To type records 5 to 1⌀ and 1⌀⌀ to 2⌀⌀ and record
8 type the following:-
    TYPEⱠ5-1⌀,1⌀⌀-2⌀⌀,8 ⌀
If the listing is to be aborted put data switch zero
on. When the program types out
        LISTING INTERRUPTED
followed by
        READY
put the data switch off.

5) DELETEⱠrec1-rec2,rec3-rec4,...recn-recn⌀ the indicated
records (same format as for TYPE option) are deleted
from the file by negating the first word of each record.
Negative numbers will become positive again. Zero
numbers will become -1.

6) ALTERⱠrecno,a,b,c,d,e,....z⌀ the record recno is
changed so as to contain the values a,b,c, etc. All
values are in free format.
e.g. To alter record 5 to contain the values.
        195,144⌀,86,-5,1⌀2 ⌀
type
        ALTERⱠ5,195,144⌀,86,-5,1⌀2

7) LISTⱠrec1-rec2, rec3-rec4.........recn-recn⌀
Same format as type option, listing may be interrupted
in same way.

8) EXIT

Notes:

No record which is beyond the HTOP value in the header
can be altered in any way. No blanks may appear in a
command unless specified. Total filed width of the
command may not exceed 79 characters.

Programmer

W. Strudwick

N.I.O. Program   266

Program Title: (For Lib. List)	Corrections to depths in GEOF files (pre Cruise 43)
Classification Letter:	S
Program Name:	COD
Version Number:	20
Type of Program:	Non-Process
Operating System:	MPX
Language:	Fortran IV
Programmer/date:	C. Spackman September 1972
Program Passed By:	C. Sapckman
Description:	COD copies information from corrected depth file onto the relevant position on GEOF
System Prerequisites:	
Inskel Common Variables:	
Subroutines Called:	JFREC, JFND, DFT
Programs Called:	
Files Called:	User Option

Operation and Method:

Program COD will transfer corrected depth readings from the corrected depth file to the GEOF file. It should only be used for cruises up to cruise 42. After this the record length of the corrected depth files was altered to 9 words and program CODFI (209) should be used.

N.I.O. PROGRAM 270

Program Title: To course update navigation on DPC1 files (off-line)
(For Lib. List)
Classification Letter: S
Program Name: UPDEP
Version Number: 20
Type of Program: Non-Process
Operating System: MPX/TSX
Language: Fortran IV
Programmer/Date: N. Olliff/June 1972
Program Passed By: C. Spackman
Description: To update the DPC1 files after navigation has been
                           changed.
System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called: GEOFn, DPC1n

Operation and Method: The program searches the relevant updated GEOF file for
starting time, finds the nearest time and extrapolates position from GEOF and
reads it to the DPC1 file. To execute , punch the following cards:-

```
//%JOB              X
//%XEQ%UPDEP
*FILES(1,GEOFn,2),(2,DPC1n,2)
*CCEND
IREC,JREC,KREC     FORMAT 3I5
```

Where IREC = first record to be updated in DPC1
      JREC = last   "   " "   "   "   "
      KREC = starting record in GEOF which must be in time before IREC

This may be repeated by further blocks of data to be updated and program is
terminated by setting IREC, JREC, and KREC to %9999%9999%9999.

The output is a listing of record number, day, time position (in latitude and
longitude) and depth (in fathoms, corrected fathoms and corrected metres).

N.I.O. PROGRAM 277

Program Title: (For Lib. List)	To check for time gaps in DATF files
Classification Letter:	S
Program Name:	DATCK
Version Number:	20
Type of Program:	Nonprocess Program
Operating System:	MPX/TSX
Language:	1800 Fortran IV
Programmer/date:	N. J. Olliff
Program Passed By:	C. Spackman

Description:

System Prerequisites:

Inskel Common Variables:

Subroutines Called:

Programs Called:

Files Called: DATF

Operation and Method:

Data in DATF files is sampled every two minutes. This program checks the time intervals, if a gap of three minutes and over is found a message is printed to that effect.

Job Description   24

```
                    //ⵄJOB           X
                    //ⵄXEQⵄDATCK
                    *FILES(1,DATFn,2)
                    *CCEND
```

Input   Data card containing JREC, KREC, IZ in format 2I5,I2

where   JREC = number of first record in data file

         KREC = "   " last   "   "   "   "

         IZ   = ∅ for data sampled every two mins.

Output   To line printer when gap detected.

         THERE IS A DATA GAP OF XX.XX MINS

         BETWEEN (Record number and contents)

         AND   ("   "   "   "   )

         When prog. terminates

         DATA GAP CHECK HAS BEEN COMPLETED BETWEEN REC XXXXX AND XXXXX

Restrictions Limited to DATF files continuing 15 words per record with time in second word and in which data is regularly sampled every two minutes.