NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

Special Purpose Utility Programs for use with the I. B. M. 1800 Computer

N.I.O. Internal Report No. N.18

AUGUST 1970

NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

SPECIAL PURPOSE UTILITY PROGRAMS

N.I.O. Internal Report No.N18.

AUGUST 1970

N.I.O.PROGRAMS 18

1) PROGRAMS

64	Plot Data Files with Records flagged by Day Number and Time	PLONK
146	Dump M Files to Paper Tape	MDUMP
147	Reconstruct M Files	MSAVE
152	Punch an IPL Paper Tape	PIPL2
161	1130/1800 Object Card to Object Paper Tape Conversion	OBJPN
162	EBCDIC Card Code to TSXPT HOLPT/KOLPT Format Paper Tape Conversion	HOLPT
172	Location and Modification on-line of Core-loads on Disk	WOTFL
197	Tape Conversion Utility	DIMCN

2) SUB-PROGRAMS

-43	Variable Format: Reads a Format Card	FMTRD
-44	Variable Format: Print I/O Buffer	PRNTB
-45	Variable Format: Read a Data Card	DATRD
-45A	Variable Format: Read a Data Card in Extended Precision	DATRX
-49	Disk Work Storage Read and Write	DREAD,DRITE
-50	Modify Core-load Define File Table	DFT
-51	Flet Look-Up	FLETM
-52	On-line Disk Patch Routine	SUBST
-54	Sector Dump Routine to Core	DKDMP
-62	Modify Define File Statement	DFTCH
-71	To Raise or Lower Plotter Pen at its Present Position	RAISE
-75 and -76	Disable and Re-enable CARDN//Test	DSLSH, RSLSH
-80	Convert Time in Tenths of a Minute to an Integer Word containing hours and minutes	KTIME
-81	Convert Time in Hours and Minutes to Tenths of a Minute	KMIN

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 ~~16,384~~ 24,576 words of core storage

3 1810

~~2~~ 2310 Disk Drives Model A

~~1~~ 2 2401 Magnetic Tape Drives (30 Kc/s) (7 Track)

1442 Model 6 Card Read - Punch

1443 Printer, 240 lines/minute

1816 Keyboard-Printer

Facit Paper Tape Reader, 1000 Characters/second

Facit Paper Tape Punch, 150 characters/second

The operating system was TSX Version 3

N.I.O.Program 64

Title Plot data files with records flagged by day number and time.

Name PLONK linked with PLONL,PLONM, PLONN, PLOND, PLONE

Machine IBM 1800

Operating System TS X

Language 1800 Fortran IV

Use File name starting and finishing times are entered on the keyboard. Code numbers for the variable to be plotted and the X - scale, together with the word position in the file record are then entered. The smallest graph is 4" high (code 1), the largest is 28" (code 7). Several graphs may be plotted one above the other, the lowest being the first one entered. The total height, allowing for 1" between graphs must not exceed the plotter width.

The VIMO codes are

01	corrected depth	06	the wind direction
02	gravity anomaly	07	solar radiation
03	Magnetic anomaly	08	air temperature
04	Course	09	relative humidity
05	true windspeed	10	ship's speed

The x - axis codes and scale are finally entered
The codes are 1) time (hours per inch)
2) distance (miles per inch)
3) distance (miles per inch),corrected for mercator projection.

Note 1 in 10^6 ≈ 13.71 n.m. per inch.

Subprograms called KEYBD,RAISE,DFT,DFTCH,PTEST,FINDR-(both in PLONK) ACORR (in PLONM)

Method

PLONK is entered once at start of program and performs the keyboard I/θ and also finds the record numbers for the entered starting and finishing times.

Calls LINK to PLONL

PLONL is entered at start of each graph. It plots the Y-scale with annotation, including starting day number and time.

Calls LINK to PLONM

PLONM reads records from the data file and sorts the required data to an array in COMMON.
Calls LINK to PLONN

PLONN plots the data from the array correcting for latitude if necessary. Half hour marks are inserted along the axis. If a day change is encountered PLOND is called which plots a Y-scale without annotation, but with the new day number .

Calls PLONM if graph not complete
PLONL if another graph is to be started
EXIT if end.

PLONE is an error program called by any of the above. Reference to the listings will be necessary to find the error condition from the number which is printed (KERR(2)).
Calls EXIT

Programmer J. Sherwood

N.I.O.PROGRAM 146 CLASSIFICATION S

Title	Dump M files to paper-tape
Name	MDUMP
Language	1800 Fortran IV
Machine	IBM 1800
Purpose	To dump the M files to paper-tape in readable format. For use in the reconstruction of the files.
Inputs	None
Output	The M files on paper tape. Name of the first followed by record then file.
Use	The program is stored in the fixed area on disk 22 on drive 0. To use:-

```
//bJOB
//bXEQbMDUMPbbbFX
```

Subroutines called	DFT
Programmer	William Strudwick.

N.I.O.PROGRAM 147 CLASSIFICATION S

Title Reconstruct M files

Name MSAVE

Language 1800 Fortran IV

Machine IBM 1800

Purpose To read the output from the program MDUMP and write it to M files on disk.

Input The output from MDUMP

Output None

Use The program is stored in the fixed area on disk 22 drive 0, to use:-

```
//bJOB
//bXEQbMSAVEbbbFX
```

Subroutines called DFT

Programmer William Strudwick.

N.I.O.Program 152 Classification S

Title Punch an IPL paper tape

Name PIPL2

Machine IBM 1800

Language 1800 Assembler (Punched as an IPL card or tape)

Operating System PIPL2 is a stand-alone program

Purpose PIPL2 allows a program in core to be punched from core location zero to 4282 (Max) for later loading as an IPL paper tape.

To use

1) Turn on Write Storage Protect Switch
2) Clear Core store
3) IPL PIPL2
4) At Wait 1, turn off Write Storage Protect Switch
5) Clear core store (This will not clear the program area)
6) Enter into core, by any available means, the program to be punched)
7) Turn MODE switch to LOAD and press RESET
8) Enter /3000 on the data switches and press LOAD I
9) Enter the number of words of the program to be punched (/2FFA max.) on the data switches and turn MODE switch to RUN
10) Ensure punch is ready and run out deleted
11) Press START
12) For more copies return to Step 7

Programmer Bernard Spatz

N.I.O.Program 161

Title — 1130/1800 object card to object paper tape conversion

Name — OBJPN

Machine — IBM 1800

Language — 1800 Assembler

Operating System — TSX

Purpose — OBJPN reads 1130/1800 card object decks and punches them on paper tape in TSXPT OBJPN format. These tapes can be read by TSXPT, 1130 Disk Monitor 1and 2, 1130/1800 card/P.T.system and by C.E. diagnostic readers.

To use — Ensure paper tape punch is ready and run out blank tape.

```
// JOB
// XEQ OBJPN
*CCEND
```

followed immediately by the card object deck, followed by a terminator card with all rows punched in column 80.

Subroutines required — CARDN, PAPTN

Programmer — Bernard Spatz.

N.I.O. Program 162 Classification S

Title EBCDIC card code to TSXPT HOLPT/KOLPT format paper tape conversion.

Name HOLPT

Machine IBM 1800

Language 1800 Assembler

Operating System TSX (Version 3)

Purpose HOLPT punches HOLPT/KOLPT format paper tape from EBCDIC source cards, for use as skeleton builder control tapes under the TSXPT operating system.

To use Ensure paper tape punch is ready and run out blank tape

```
// JOB
// XEQ HOLPT
*CCEND
```

Followed immediately by the skeleton builder control cards as described in "TSX operating procedures". Follow with a terminator card with all rows punched in column 80.

Subroutines required CARDN, PAPTN

Programmer Bernard Spatz.

N.I.O. PROGRAM 172

Title — Location and modification on-line of core-loads on disk.

Name — WOTFL

Machine — IBM 1800

Language — Fortran IV

Purpose — By use of console 1816 typewriter to seek out a core-load or data file on disk and optionally dump it or modify it.

Input — Is from keyboard in response to message on typewriter, after initial Job and XEQ cards.

Job Description

```
// JOB                 X
// XEQbWOTFLbbbFX
```

Required Subroutines

FLETM, DKDMP, SUBST

Data Format

1st record - name of the file in 3A2 format left justified NAMFLb

2nd record - a flag in I1 format

- Flag 1 find a new file with name specified on next record in 3A2 format.
- Flag 2 dump data between specified word limits relative to start of last named file
- Flag 3 modify a word in the last named file
- Flag 4 exit from WOTFL

3rd record - dependent on flag

- Flag 1 as for record 1
- Flag 2 1st and last words (2I6 format), relative to start of last named file, to be dumped to list printer. Integer 0 for Hex and 1 for decimal dump in I2 form
- Flag 3 relative address in last named file and new value in decimal (2I6).

Note: 1. Records 2 and 3 may be repeated until flag 4 is read.

2. Remember typewriter input must fill field specified in format and therefore some fitting out of some entries with spaces may be required

3. List printer on ship is the typewriter.
4. Relative addresses start at 1 not zero.
5. The current flag option may be aborted for flags 2 and 3 by entering 1 in column 1 and 12 spaces.

Output

1. After initial entry and flag 1 - the type of file requested i.e. combination, interrupt, mainline, non-process or data file. Then the disk sector location (decimal) and sector count (for data files and entries) or word count (for core loads and / entries).

2. After flag 2 - a dump to list printer of the specified area on disk. This dump may overlap file boundaries. The TSX dump routine is used and this gives one line of extraneous infomation first.

3. The old value of the word altered and the new value. The relative address of the requested word is in this case checked to confirm that it is within the named file.

Restrictions None

Execution Program operates in conversational mode.

Programmer James Crease.

N.I.O.PROGRAM 172

Amendment

Output 2. should read:-

After flag 2 - a dump to list printer of the specified area on disk. This dump may overlap file boundaries. The TSX dump routine is used and this gives one line of extraneous information first, which should be ignored, the first column of numbers are core locations used in the dump routine and are irrelevant.

N.I.O. PROGRAM 197

Title	Tape Conversion Utility
Name	DIMCN
Machine	IBM 1800
Language	1800 Assembler
Purpose	To convert OBJPN tapes by removal of excess delete character.
Operation	Initial Program Load DIMCN (after clearing core). It waits with /3003 in B register.
Input	Place the OBJPN produced tape in the reader, with the head just before the first non-delete character on the tape. (The first non delete character will be /36, i.e. the word count 54.) Press Start. Press stop when input tape finished.
Output	The converted tape will be punched out.
Programmer	Bernard Spatz.

N.I.O. SUBPROGRAM -43

Title

Variable Format: Reads a format card.

Name

Subroutine FMTRD

Machine

IBM 1800

Language

1800 Assembler

Purpose

FMTRD reads one card containing a format and stores it in a form suitable for the subroutine DATRD (N.I.O. SUBPROGRAM -45).

Use

Calling sequence:

```
CALL FMTRD (FORMT,ERROR)
```

FORMT must be an integer vector fifty words long. ERROR is an integer word. Upon return, FORMT contains the translated format and ERROR will be zero. If an error was detected ERROR will contain the next column to be processed. When an error is detected no attempt is made to complete the translation and the format may have to be changed.

Format codes: The following specifications are acceptable:

WX nIW nFW.d nEW.d

n may be omitted if it is one. One level of parenthesis is allowed for group repetition. In addition, parentheses are required around the entire format. Every specification, including WX and parenthesized groups, must be followed by either a comma or a right parenthesis. Multiple record formats (/), scaling (P) and alphabetic conversion (A,H) are not available. In addition, the format must be completed on one card.

Subroutines Used

CARDN, HOLEB, SBUF.

Notes

This is based on EPL subroutine FMTRD which is part of the 1130 Statistical System (1130-CA-06X).

Restrictions

The subroutine SBUF contains the I/O buffer and is used by subroutines PRNTB and DATRD. Thus, it must never be localized.

Programmer

M. Fasham.

N.I.O. SUBPROGRAM -44

Title Variable Format: Print I/O Buffer.

Name Subroutine PRNTB

Machine IBM 1800

Language 1800 Assembler

Purpose PRNTB prints the I/O buffer contained in subroutine SBUF.

Use Calling sequence:

```
   CALL PRNTB
```

When called, PRNTB prints the first eighty positions of the I/O buffer on the printer. It may be used after a call to FMTRD or DATRD whether or not an error occurred, to print the card just processed.

Subroutines Used EBPRT, PRNTN, SBUF

Restrictions The subroutine SBUF must never appear in a local statement.

Notes Based on EPL subroutine PRNTB.

Programmer M. Fasham.

N.I.O. SUBPROGRAM -45

Title — Variable Format: Read a Data Card

Name — Subroutine DATRD

Machine — IBM 1800

Language — 1800 Assembler

Purpose — DATRD reads one card of data according to a format previously stored by FMTRD.

Use — Calling sequence: CALL DATRD (FORMT, ERROR, VAR1, N1, VAR2, N2, ... , φ, φ).

FORMT is an integer vector fifty words long previously named in a call to FMTRD. ERROR is an integer word. VAR1, VAR2, etc. are integer or real variables or vectors. N1, N2 etc. are integer variables or constants. Each is positive if the corresponding variable is integer negative if real.

Upon return, the first Ni locations of each VARi are replaced by data. Automatic type conversion from I specification to real and from E and F specification to integers is performed. If no error is detected, ERROR is set to zero· otherwise it is set to the next column to be processed. None of the Ni may be zero. Two zeros ends the list of variables.

Data — Only one data card, at a time can be read by this routine. An attempt to read beyond the end of the format is treated as an error. Numbers may have any number of leading or trailing blanks. Signs may have leading and trailing blanks. If the sign is omitted, it is assumed to be positive. For F and E conversions, a decimal point is allowed· if omitted it is implied by the format. E type numbers may have an exponent part which must start with an E, a blank or a sign. Blanks may not precede the E. If the exponent minus the number of decimals (explicit or implicit) is not in the range ±63, an error is indicated. If the absolute value of the number ignoring the decimal point and exponent is greater than $2^{31} - 1$, the result will be incorrect with no error indication given. An overflow or underflow condition is possible and is ignored.

Subroutines Used — CARDN, HOLEB, SBUF, NORM, GMPYX, GDIVX, IFIX, FSTOX.

Notes — This subroutine is based on one which is part of the EPL Statistical System 1130-CA-06X.

Restrictions — The subroutine SBUF must never be localised.

Programmer — M. Fasham.

N.I.O. SUBPROGRAM -45/A

Title	Variable Format: Read a data card in extended precision.
Name	Subroutine DATRX
Machine	IBM 1800
Language	1800 Assembler
Purpose	DATRX reads one card of data according to a format previously stored by FMTRD. DATRX differs from DATRD in that real variables are stored in extended precision.
Use	Same as DATRD (N.I.O. Subprogram -45).
Data	Same as DATRD.
Subroutines Used	CARDN, HOLEB, SBUF, NORM, GMPYX, GDIVX, IFIX, ESTOX.
Restrictions	Same as DATRD.
Programmer	M. Fasham.

N.I.O. SUBPROGRAM -49

Title

Disk work storage read and write

Name

Subroutine DREAD and DRITE

Language

1800 Assembler

Machine

IBM 1800

Purpose

To read or write an integer array of 320 words to a specified sector in either process or non-process working storage (depending on type of main program) on a specified disk drive.

Method

(a) Called by CALL DREAD(IDISK,ISECT,IFILE) to read, where

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

ISECT is the relative sector number in working storage (≥0),

IFILE is the array for data to be read into. IFILE must be dimensioned IFILE(322); IFILE(320) will contain the first word of the sector; IFILE(1) will contain the last word.

IFILE(321) and (322) are used by DREAD.

(b) Called by CALL DRITE(IDISK,ISECT,IFILE) to write, where the arguments are as for (a); IFILE(1) must contain the last word of the sector and IFILE(320) must contain the first word.

An *IOCS(DISK) record is not required.

Restrictions

A complete sector (320 words) of data must be read or written. Loss of data will result if this is not adhered to.

If the specified sector is greater than the number of sectors allocated for work storage then TASK error 0026 or SYDIR EAC I13 will result.

Subroutines used

DISKN, TVSAV, TVEXT.

Execution time

Depends on the position of disk arm at time of call, not more than 1·4 seconds, typically 0·1 second.

Core requirements

58 words (whereas FORTRAN disk I/O routine MDFIO uses 883 words).

Programmer

D. Brown

N.I.O. SUBPROGRAM -50

Title

Modify Core-Load Define File Table

Name

DFT

Machine

1800

Language

Assembler

Purpose

To modify at run-time the in-core define file table of a core-load. The intention is that a core-load which references a number of different files at different times in an identical way may be built initially with *files referencing only one file. In fact it may be built without a *FILES card at all. (i.e. referencing work storage.)

Input

Calling sequence is

```
    CALL DFT(N,NAMFL,IERR)
```

N must equal the number of definedfiles (FORTRAN DEFINE FILE) in the original core-load.

NAMFL is a single precision integer array dimensional 3N in which the names of N data files must be placed in 3A2 (left justified) format before the CALL.

Output

IERR is set as follows

0 No errors in DFT

1 N does not match number of definedfiles

2 Name of file cannot be found on drives searched (see below)

3 The file requested is not a data file.

Note: IERR must be checked by the calling program to make sure no errors have occurred.

The DFT table 6 word entries on completion of the call have been modified in the following respects.

Word 1 - logical file number is as in the original DEFINE FILE statement

Word 2 - Number of records in file - calculated by DFT from word 6 and number of sectors in file

Word 3 - Record length in words - as in original DEFINE FILE statement

Word 4 - Address of associated variable - as originally compiled

Word 5 - Drive code and address of file named in NAMFL array

Word 6 - Number of records/sector - as calculated from word 3.

<u>Notes</u> 1. Word 2 is set to /7FFF if the calculation gives a greater number than this.

2. Define file table entries are set up in the order in which the original DEFINE FILES statements were made. The first name in NAMFL is associated with the first defined file and so on.

3. If there is an error the table is unaltered.

<u>Required Subroutines</u> FLETM

<u>Method</u> The address of the table is found from the CDW entry in the core-load. Subroutine FLETM is called to search the disk drive FLET tables to match the name entered in NAMFL with a data file. If the calling program is a process one all drives on the system are searched. If it is non-process only those drives in use for the JOB are searched. If a match is found then the DF table is modified as indicated above by using the information in the FLET entry.

<u>Programmer</u> J. Crease

AMENDMENT

Subprogram -50

Title Modify Core-Load Define File Table

Name DFT

The use of name 9WORK in the files names in array NAMFL allows DFT to skip over the DFT entry. Thus if 9WORK is the 3rd name in the NAMFL list the 3rd defined file will not be modified. A particular need for this arises if working storage files have been defined and are required for use by a program.

e.g. if files 1 and 2 are work storage and
file 3 is fixed area

then if NAMFL has the names 9WORK, 9WORK, DATA the Define file table will leave 1 and 2 as they were (i.e. work storage) and set file 3 to file DATA.

N.I.O. SUBPROGRAM -51

Title Flet look up

Name FLETM

Machine 1800

Language 1800 TSX

Purpose To search Flet on specified drive for a specified core-load or data file.

Calling Sequence CALL FLETM(NAME,IDRIV,ISEC,IWD,ITYPE).

Inputs NAME contains the name NAMEPb of the core-load
or data file or system area to be looked for.
NAMEPb is in 3A2 format left justified so
NAME must be dimensioned 3.
IDRIV is the number of the logical drive searched.

Outputs ISEC drive code and sector address if found,
otherwise O

IWD word count for core-loads and sector
count for data files

ITYPE O for data file
1 for non-process core-load
2 for mainline core-load
3 for interrupt core-load
4 for combination core-load.

Restrictions Flet will not be searched if the drive
requested is not in use for the JOB.
If the call is from a process core load
it is up to calling program to check that
drive is on the system

Programmer J. Crease

N.I.O.SUBPROGRAM -52 CLASSIFICATION S

Title On-line disk patch routine

Name SUBST

Language IBM 1800 Assembler

Purpose To substitute a new word from core in a specified word on disk

Call CALL SUBST(IWRD,ISEC,IVAL)

where

IWRD contains word position in sector of word to be altered

ISEC contains sector address (with drive no. in first hex position)

IVAL contains the new number to be placed in the above address.

The original value in the disk address is returned to the calling program in IVAL

Notes

1. This routine will write to any sector file protected or not. It is up to the user to verify that if he is modifying a fixed area file that he is not writing beyond the file boundaries. As an example Program WOTFL does this sort of test.
2. IOCS cards are not used for this routine

Space 60 words and 321 words core buffer for disk sector

Programmer J. Crease.

N.I.O.SUBPROGRAM -54 CLASSIFICATION S

Title Sector Dump Routine to Core

Name DKDMP

Language 1800 Assembler

Purpose To dump a sector of disk into core

Call CALL DKDMP (IDMP(322),L)

where

1. IDMP is a Fortran array dimensioned 322 in the calling program and IDMP(321) contains the drive number (first hex digit) and sector address (hex digits 2-4) of the sector to be dumped whilst IDMP(322) contains the word count of 321.

2. L has in it on return from the call the absolute core address of IDMP(1).

Note IOCS cards are not required for this routine.

Space 36 words

Programmer J. Crease.

N.I.O.SUBPROGRAM -62

Title   Modify define file statement

Name   DFTCH

Machine   1800

Language   Assembler

Purpose   To modify the in-core define file table of a core-load so that the number of records and the lenght of each record in a disk file may be altered at run-time.

Input   The calling sequence is

CALL DFTCH (LOGFL,NAMEF,IDRIV,NOREC,IRECL,IERAD)

where LOGFL is the logical file number

NAMEF is the file name

DRIV is the number of the drive which contains the data file

NOREC is the requested number of records in the file

RECL is the requested record length in words.

Output   ERAD is set as follows:-

0   No error

1   Record length greater than 320 words

2   Requested file length greater than existing file length

3   File number is not in DFT

4   File name in calling sequence is not the name in the DFT

5   File is not on requested drive

N.B IERAD must be checked by the calling program to ensure that no errors have occurred.

The DFT table, modified as follows:-

Word 1 - logical file number as in the original DEFINE FILE statement

Word 2 - number of records in the file as specified by the calling program

Word 3 - record length in words as specified by the calling program

Word 4 - address of associated variable - as originally compiled

Word 5 - drive code and address of file

Word 6 - number of records/sector - calculated from word 3

N.B. If there is an error the table is unaltered.

Required subroutines FLETM

Method The program tests for greater record length than 320 words and for a requested file length greater than the existing file length. If an error occurs the DFT table is unaltered and an error parameter is set up. FLETM is used to calculate the number of records in a sector. The start of the DFT is accessed and each table is searched until the correct file is found. If the file is not found an error is set, otherwise words 2, 3 and 6 are modified and the subroutine restores the accumulator and extension before returning to the calling program.

Restrictions Only one DEFINE FILE statement is modified by each "CALL". The data file must be in fixed area. All numerical parameters must be integers.

Programmers Eileen Page, Jackie Webster.

N.I.O. SUBPROGRAM -71

Title To raise or lower plotter pen at its present position.

Name RAISE

Machine 1800

Language 1800 Assembler

Purpose Avoids calling EPLOT to raise or lower pen.

Use CALL RAISE raises pen at present position.
CALL LOWER lowers pen at present position.

Subroutines Used PLOTX

Programmer J. Sherwood

N.I.O. Subprograms -75 and -76 Classification U

Title	Disable and re-enable CARDN // test
Name	Subroutines DSLSH and RSLSH
Machine	IBM 1800
Language	1800 Assembler

Usage

CALL DSLSH will disable the check for // cards in CARDN. This is often useful when processing source programs as data for a documentation program.

CALL RSLSH will re-enable the CARDN // check and would be used after all reading of any likely // cards within the program. If this is not done, a TASK cold start will be needed before the system is usable again.

Method

DSLSH sets word 32 of the current level work area non-zero and RSLSH sets word 32 back to zero.

Programmer

David Brown.

N.I.O.SUBPROGRAM -80 CLASSIFICATION S

Title Convert time in tenths of a minute to an integer word containing hours and minutes.

Name Function KTIME

Machine 1800

Language Fortran IV

Purpose All times used for input or output on the ship system must be in the form of one 4 digit integer word (HHMM where HH is hours and MM minutes). However all times stored on the dynamic data files are in tenths of a minute and KTIME can be used to convert these times to the hours and minutes format.

Use I = KTIME(J)

where,

I is an integer containing the time in the form HHMM
J is an integer containing the time in 1/10 minute.

KTIME rounds upwards to the nearest minute by adding 5 to J before conversion.

Programmer M. Fasham.

N.I.O. SUBPROGRAM -81 CLASSIFICATION S

Title	Convert time in hours and minutes to tenths of a minute
Name	Function KMIN
Machine	1800
Language	Fortran IV
Purpose	This performs the reverse function to KTIME
Use	J = KMIN (I) where, J is an integer containing the time in 1/10 minute I is an integer containing the time in the form HHMM.
Programmer	M. Fasham