

**JAERI-M**  
**91-201**

**DATA-POOL : A DIRECT-ACCESS DATA BASE  
FOR LARGE-SCALE NUCLEAR CODES**

December 1991

Naoki YAMANO\*, Kinji KOYAMA, Yoshitaka NAITO  
and Kazuyoshi MINAMI\*\*

**日本原子力研究所**  
**Japan Atomic Energy Research Institute**

JAERI-M レポートは、日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合せは、日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）  
あて、お申しこしください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11茨城  
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department  
of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun,  
Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1991

---

編集兼発行 日本原子力研究所  
印 刷 日立高速印刷株式会社

DATA-POOL : A Direct-access Data Base  
for Large-scale Nuclear Codes

Naoki YAMANO\*, Kinji KOYAMA, Yoshitaka NAITO  
and Kazuyoshi MINAMI\*\*

Department of Fuel Safety Research  
Tokai Research Establishment  
Japan Atomic Energy Research Institute  
Tokai-mura, Naka-gun, Ibaraki-ken

(Received November 5, 1991)

A direct-access data base DATA-POOL has been developed for large-scale nuclear codes. The data can be stored and retrieved with specifications of simple node names, by using the DATA-POOL access package written in the FORTRAN 77 language. A management utility POOL for the DATA-POOL is also provided. A typical application of the DATA-POOL is shown to the RADHEAT-V4 code system developed for performing safety analyses of radiation shielding. Many samples and error messages are also noted to apply the DATA-POOL for the other code systems.

This report is provided for a manual of DATA-POOL.

Keywords : DATA-pool, POOL, Computer Code, Software Package, Direct-access, Data Base, Data Handling, RADHEAT-V4, VISUAL, Nuclear Code, Manual

---

\* Sumitomo Atomic Energy Ind., Ltd.

\*\* Fujitsu Limited.

DATA-POOL：大規模原子力コード用直接編成データベース

日本原子力研究所東海研究所燃料安全工学部

山野 直樹<sup>\*</sup>・小山 謙二・内藤 椹孝・南 多善<sup>\*\*</sup>

( 1991 年 11 月 5 日受理 )

直接編成ファイルを用いたデータベース DATA-POOL を大型原子力コードのために開発した。データは簡単なノード名の指定によって格納・検索される。DATA-POOL 処理パッケージは FORTRAN 77 言語で作成されている。保守管理ユーティリティ POOL も併せて用意されている。DATA-POOL の典型的な応用例として、放射線遮蔽安全解析コードシステム RADHEAT-V4 への適用を示した。DATA-POOL を他のシステムに適用する為の多くの使用例及びエラーメッセージについても述べている。本報告書は DATA-POOL の使用手引書である。

---

東海研究所：〒319-11 茨城県那珂郡東海村白方字白根 2-4

\* 住友原子力工業(株)

\*\* 富士通(株)

## Contents

1. Introduction .....	1
2. Structure of DATA-POOL .....	3
3. Function of DATA-POOL .....	11
4. Access Method for DATA-POOL .....	26
5. TSS Management Utility of DATA-POOL .....	44
6. Record Format of DATA-POOL for RADHEAT-V4 .....	69
7. Concluding Remarks .....	93
Acknowledgements .....	93
References .....	93
Appendix A Error Message of DATA-POOL .....	95
Appendix B Program List of DATA-POOL Access Package .....	99
Appendix C Program List of POOL .....	141

## 目 次

1. 序 言 .....	1
2. DATA-POOLの構造 .....	3
3. DATA-POOLの機能 .....	11
4. データプール入出力法 .....	26
5. TSS会話型データプール管理ユーティリティ .....	44
6. RADHEAT-V4におけるデータプール記録型式 .....	69
7. 結 び .....	93
謝 辞 .....	93
参考文献 .....	93
附録A データプールエラーメッセージ .....	95
附録B データプールアクセスパッケージプログラムリスト .....	99
附録C POOLプログラムリスト .....	141

## 1. Introduction

For estimating radiation damage and dose rate in shielding safety evaluation, precise calculations have been performed by using large-size multi-group transport codes according to progress of computer resource. An amount of data on cross sections and radiation distributions becomes so large and the structure of them becomes also so complex that it is not easy to treat them using the conventional sequential data form. To avoid the difficulty and execute the effective processing, a direct-access data base DATA-POOL has been developed for the large-size nuclear codes.

The basic concept of DATA-POOL was designed by the JAERI Nuclear Code Committee. A DATAPOOL code<sup>1)</sup> was developed at the JAERI Computing Center in 1980. This DATAPOOL is designed for an general data base that is possible to store many node name's and contains an intelligent algorithm for information retrieval by using a LRU (Least Recent Use) table and a DT (Directory Table) buffer. Arbitrary relations of node names can be defined by the user. The DATAPOOL code, however, has a pre-compilation process to translate the CJ statement, which is similar to the FORTRAN language, to the ordinary FORTRAN statements and the program debugging is often trouble some.

As for shielding calculation, a large amount of data is required, however the number of classified node name is not so great. Therefore, the direct-access data base with the large directory retrieval is not necessary to treat the cross sections or radiation distributions. Present DATA-POOL access package which is different from the DATAPOOL code is therefore developed to treat effectively the cross sections and the other related data.

The DATA-POOL access package is characterized as a simplified version of the DATAPOOL code and reduces I/O access times by half compared with the DATAPOOL code to retrieve the same data due to a simple and an appropriate forms of directories. The minimum procedure is required to find the specified node name and the data. The DATA-POOL access package is suitable to treat data of large size with a node name.

The access package consists of several subroutines written by the FORTRAN 77 language. An ASSEMBLER subroutine is only used to get DCB (Data Control Block) information in the DD (Data Definition) statement. The user can use the direct-access data base to assign the load module of the DATA-POOL access package as a private library for the linkage editor.

The DATA-POOL has been adopted to the RADHEAT-V4<sup>3)</sup> code system developed for shielding safety evaluation. The experience of using the

access package about 5 years shows that the DATA-POOL is operated regularly. The default value of a physical value of a physical record length is set to 3600 bytes (900 words), however it can be easy to change the appropriate value according to user's system.

The data in the DATA-POOL are classified and identified by a node name consisted of 4-characters. On the other hand, the data are stored and retrieved by the node name specification. The structure of the DATA-POOL is described in Chap. 2. The access package of the DATA-POOL and the utilization are described in Chapters 3 and 4, respectively.

A management utility POOL operated with TSS terminals is prepared for the management of DATA-POOL. The utility has 14 functions such as copy, delete, condense, backup and rename. The usage of the utility POOL is described in Chap. 5. To show the application of DATA-POOL, the special data forms used in the RADHEAT-V4 code system are described in Chap. 6. The data in the DATA-POOL can be plotted as two-, three-dimensional graphs and contour-line maps by using a plotting utility VISUAL<sup>4)</sup>.

The error messages printed by the access subroutines are noted in Appendix A. The source lists of the access subroutines and of the utility POOL are shown in Appendices B and C, respectively.

## 2. Structure of DATA-POOL

DATA-POOL is a direct access data set defined by the direct access read/write statements of the FORTRAN 77 language. The data set has a fixed and an unblocked record length of 3600 bytes (900 words).

DATA-POOL consists of three sections named a "Control Section", a "Directory Section" and a "Data Section" as shown in Fig. 2.1. Arbitrary data are stored/retrieved in the DATA-POOL by a standardized format, so that the management of the data can be easily achieved. In this chapter, the concept and the structure of the DATA-POOL are described. The management and plotting utilities related to the DATA-POOL are briefly described in Section 2.1.

### 2.1 Concept of DATA-POOL

In the DATA-POOL, the data are labelled by an arbitrary node name which consists of 4-characters defined by the user, and stored in the Data Section. The node name is related to the others and a tree structure is generated as shown in Fig. 2.2.

In this figure, EGRP means an energy group structure commonly used in the data. SGRX means the attribute of secondary gamma-ray production cross sections. INFX means the attribute of infinite dilution cross sections. ELA means the attribute of elastic scattering matrices. The node name frequently referred in the system should be located at the upper level of the tree structure. Data belonging to the same category should be combined and attributed to a node name in order to improve the efficiency of data retrieval.

The tree structure and the node name are stored in the Directory Section together with the direct-access record addresses. The data retrieval is carried out from the node of the first level to that of the lower level, so that the most suitable tree structure is essentially needed according to the property of the data.

The data retrieval for the DATA-POOL is carried out by setting a series of the node names in PFIND/PSET subroutines, and then the data access is carried out by PREAD/PRITE subroutines.

The DATA-POOL adopts an exclusive control for the write access in order to prevent the destruction of DATA-POOL from plural job access. In the period executing between PWSTAT subroutine and PWEND subroutine, the write

access by the other jobs is inhibited. These functional subroutines are described in Chap. 3.

The management of the DATA-POOL is carried out by using an utility program POOL. POOL has 14 functions to maintain the data in the DATA-POOL such as initialize, rename, copy, backup. These functions can be executed with TSS terminals or batch jobs. The data in the DATA-POOL can be displayed on TSS graphic terminals (TEKTRONIX T4014) or NLP (Nihongo Line Printer) by using a plotting utility program of VISUAL. Various plottings such as two-dimentional, contour-line and three-dimentional graphs are produced by using the conversational input data. The relation between the utilities and the DATA-POOL is shown in Fig. 2.3.

## 2.2 Control Section

The Control Section is located at the first record of DATA-POOL and has a size of 40 words. The variables in the section are used for the control of the DATA-POOL. The record structure of the Control Section is shown in Fig. 2.4. The Control Section should be contained in a physical record length of the DATA-POOL. The initialization of the Control Section is carried out by calling PINIT subroutine.

## 2.3 Directory Section

The Directory Section takes an important role which determines the relation between the node name and the record address of the data, and has the information of the lower nodes in the tree structure. The Directory Section consists of the several sub-directories. The sub-directory has the information of a node name, a head address of the data section associated with the node name, the date of creation and control variables defined by the user. The structure of the Directory Section is shown in Fig. 2.5. A sub-directory takes 12 words in the Directory Section and a directory holds a physical record length, so that the maximum number of nodes associated with the same level is limited to the next value.

$$N_{\max} = \text{tranc} \left\{ \frac{(1 \text{ physical record length (words)} - 4)}{12} \right\}$$

The DATA-POOL of RADHEAT-V4 has a physical record length of 3600 bytes (900 words), so that the value of  $N_{\max}$  is 74.

A feature of the DATA-POOL is the information for the nodes of the lower level can be obtained at once by referring a directory, however the DATA-POOL will not be adequate for systems with tree structures contained too many lower nodes.

In the sub-directory, 5 kinds of information can be recorded by the user. The record area is prepared for the reason which the DATA-POOL consists of many kind of data associated with a node name, and will be used as control flags whether data are in existence or nonexistence in the Data Section. The needless access to the data files can be prevented by utilizing the information in the sub-directory.

#### 2.4 Data Section

The Data Section consists of several sub-data sets. A sub-data set is created by executing a writing. The writing is carried out by calling a subroutine of PRITE - PRITE4. The subroutine PRITE only creates the comments of the node. The data of one-dimensional array are written in the regions from DATA1 to DATA4 by calling a subroutine of PRITE1 - PRITE4, respectively. (see Fig. 2.6) The subroutine PRITE1 creates the comments of the node and the DATA1. The subroutine PRITE4 generates the comments of the node, DATA1, DATA2, DATA3 and DATA4 as shown in Fig. 2.6. These data can be read by using the subroutines of PREAD - PREAD4 which correspond to the PRITE - PRITE4 subroutines.

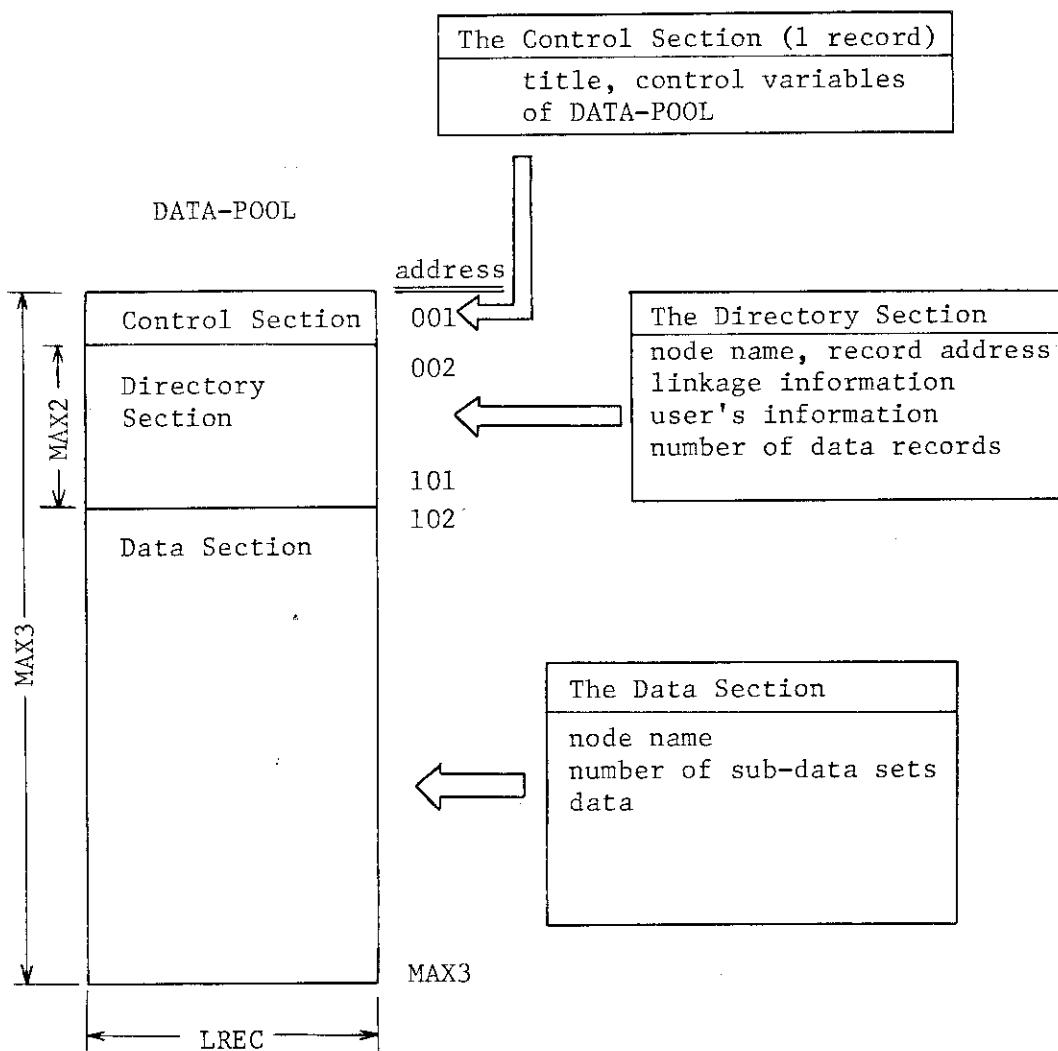


Fig. 2.1 Basic concept of DATA-POOL

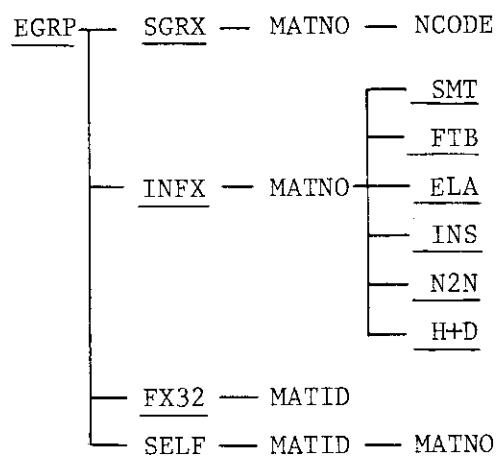


Fig. 2.2 Fundamental node tree structure

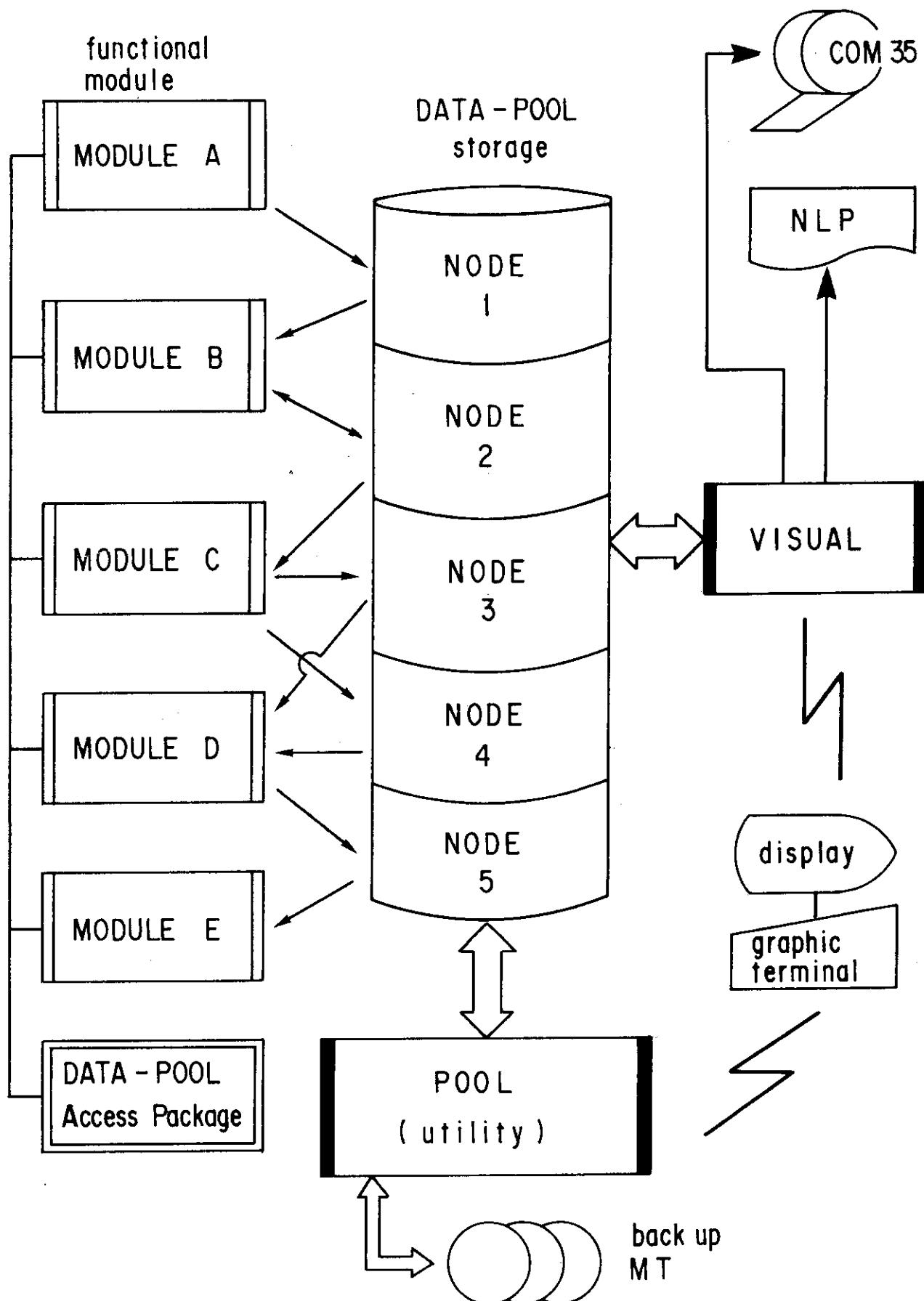


Fig. 2.3 DATA-POOL system structure

No.	Variable	Data Information
1	TITLE(1)	title of the DATA-POOL
	{	}
20	TITLE(20)	data set name, revised data, contents of the DATA-POOL et al.
21	NA1	
22	NA2	
23	NA3	
24	KEY1	address for the directory of the first level node
25	KEY2	head address of the vacant directory section
26	LREC	head address of the vacant data section
27	MAX1	write flag for the exclusive control
28	MAX2	read flag for the exclusive control (not used)
29	MAX3	length of a physical record (words)
30	NREAL1	maximum number of the same level node
31	NREAL2	size of the directory section
32	--	size of the data section
	{	number of used records in the directory section
40	--	number of used records in the data section
	{	for future use
	--	
LREC	--	dummy (not used)
	--	

Fig. 2.4 Structure of the Control Section

No.	Variable	Data Information
1	NODE	node name
2	DUMMY	for future use
3	NAUP	address of the upper node directory
4	ITEM	number of the sub-directory
5	NODES	node name of the first lower node
6	NRECS	number of physical records
7	NADWN	address for the directory of the lower node ( zero means not exist )
8	NADAT	address for the data set associated with this node ( zero means not exist )
9	NDASET	number of the sub-data set ( zero means not exist )
10	NDATE(1)	date of creation ( YY-MM-DD )
11	NDATE(2)	YY:year, MM:month, DD:day
12	INFOM(1)	information defined by the user
13	INFOM(2)	information defined by the user
14	INFOM(3)	information defined by the user
15	INFOM(4)	information defined by the user
16	INFOM(5)	information defined by the user
17	NODE	node name
18		
19		
.		
.		
.		
LREC		

Fig. 2.5 Structure of the Directory Section

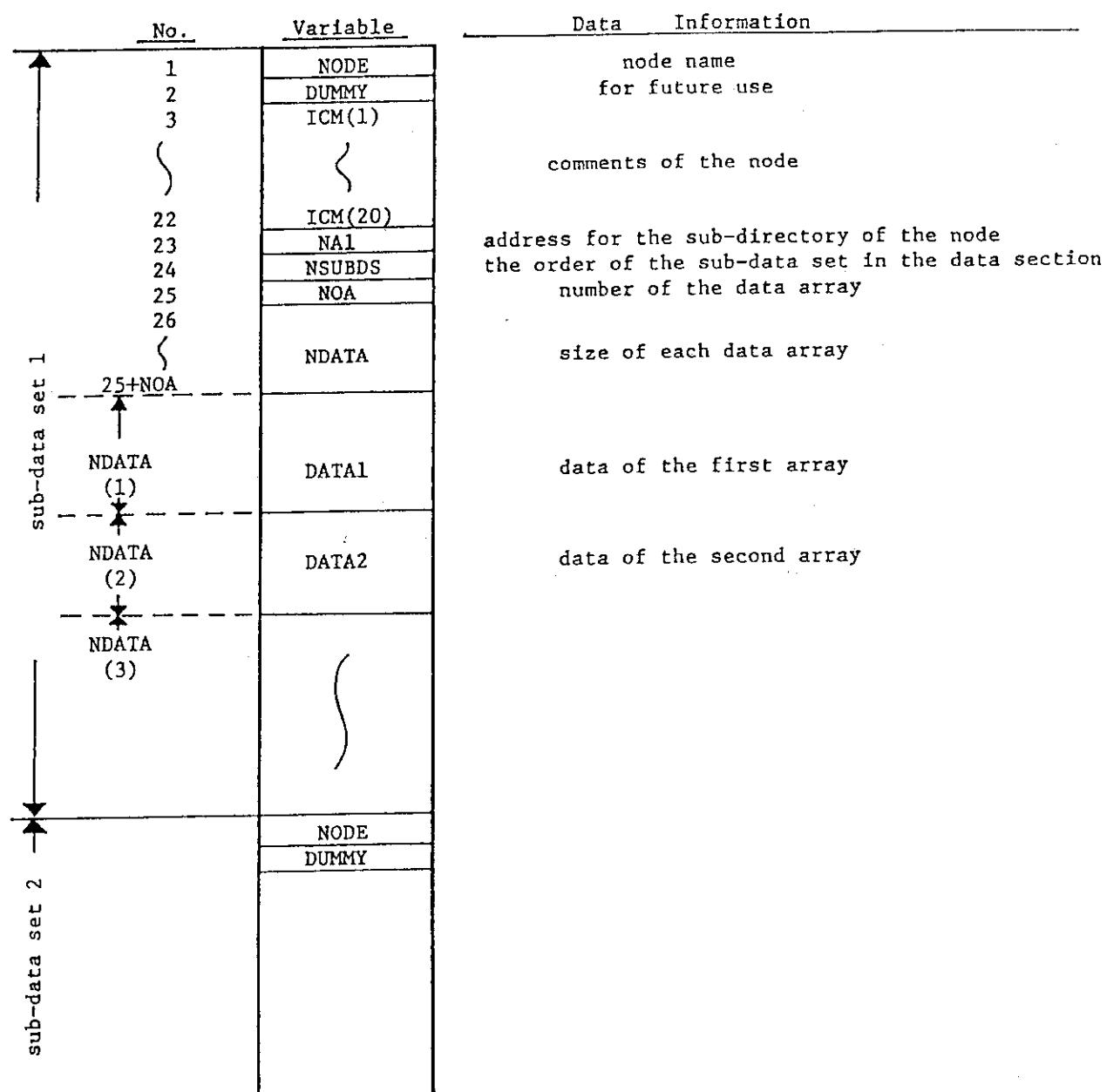


Fig. 2.6 Structure of the Data Section

### 3. Function of DATA-POOL

The access for the DATA-POOL is carried out by using access subroutines. Users can easily treat the data in the DATA-POOL to call the access subroutines in user's program written in FORTRAN77 language. An outline of the access subroutines is shown in Section 3.1 and the description of the common table is shown in Section 3.2. The variables of each access subroutine are described in Section 3.3.

#### 3.1 Access Subroutine

Access subroutines are written in FORTRAN77 language. A subroutine GETDCB is only written in ASSEMBLER language. 29 access subroutines are stored in the DATA-POOL access package and these functions are as follows:

- (1) PINIT : initialize DATA-POOL and clear the Control Section,
- (2) POPEN : declare the access of DATA-POOL and open the data set,
- (3) PWSTAT: declare the start of writing to DATA-POOL and set the exclusive control,
- (4) PWEND : declare the end of writing to DATA-POOL and reset the exclusive control,
- (5) PSET : set the node name and record address to the Directory Section in order to write the data,
- (6) PFIND : retrieve the node name and record address from the Directory Section in order to read the data,
- (7) PRITE : write data to the Data Section,  
PRITE4
- (8) PREAD : read data from the Data Section,  
PREAD4
- (9) PDELT : delete node name and data,
- (10) PAGET : retrieve next record address to be read or write access,
- (11) PDGET : retrieve the directory information,
- (12) PASTO : set record address to be read or write access,
- (13) PSKIP : skip arbitrary logical records,
- (14) SETMSG: set the maximum record size of DATA-POOL,
- (15) WRTCHK: set the number of records to be write,

- (16) NODEER: error check,
- (17) CATLST: display the Directory Section,
- (18) GETDCB: retrieve the DCB information of the data set.

The other subroutines are supplementary ones for the subroutines described above, so that the descriptions are abbreviated.

### 3.2 Common Table

The control information of DATA-POOL is set to the common area, and used by each subroutine. The initialization of the variable is carried out by using a subroutine PINIT or POPEN. The variables in five common tables are described below.

#### (a) /DPCONT/

LCONTR	:	size of control section (normally 40 is set),
NCONTR	:	maximum number of allocated files (99 is set),
ICTR(40,99)	:	informations of the Control Section,
IX	:	address variable of the direct access file,
NSUBDS	:	number of sub-data set to be write in a data set,
NDSTAT	:	start address to be write.

#### (b) /DPWORK/

LBUFFR	:	size of buffer area (1000) greater than a physical record length of DATA-POOL,
LRECOD	:	length of a physical record,
IBUFFR(1000)	:	working area for input/output access,
NRECOD	:	maximum number of physical records,
NODE1	{ }	
NODE2		
NADWN		
NADAT		information of the sub-directory,
NDASET		
NDATE(2)		
NINFOM(5)		
NUTOLD	:	logical unit number recently accessed,
NTHOLD	:	level of node recently accessed,

NODOLD(10,2) : node name and address of sub-directory recently accessed,  
 NA1 : address of directory.

## (c) /DPWCHK/

NRPEMT : number of records possible to write,  
 NRWRDN : number of physical records written in the Data Section,  
 NIXOLD : starting record address to be write,  
 NUTWTN : logical unit number to be write,  
 NTHWTN : node level to be write,  
 NODWTN(10,3) : history of directory and sub-directory for the node,  
     (1,1) ; node name  
     (1,2) ; history flag for sub-directory  
     (1,3) ; history flag for directory  
     flag : 0 = newly created  
           1 = not used  
           2 = no update  
           3 = address of lower directory in the sub-  
             directory is updated  
           4 = address of data set in the sub-directory  
             is updated  
 ISDBEF(12) : backup area of old sub-directory when the sub-directory  
 is updated.

## (d) /DPDELT/

IBUFF2(1000) : working area should be the same size of IBUFFR.

## (e) /DPEMSG/

IFLAG : flag whether an error of error No. 232 occurs or not,  
 NERNO : error No.

## 3.3 PINIT Subroutine

PINIT initializes a direct-access data set of the DATA-POOL and sets the maximum number of physical records.

calling sequence : PINIT (NUNIT, NDIRCT, LENGTH, ITITLE)

[Input] NUNIT : logical unit number (1 ~ 99),  
NDIRECT : number of directory records,  
LENGTH : physical record length (words),  
ITITLE : title of DATA-POOL (80 words).

subroutines called : SETMSG, ERRSET, DATE

### 3.4 POPEN Subroutine

POOPEN declares the use of DATA-POOL and sets DCB information of the data set. This subroutine must be called as the first one when the access for DATA-POOL is done.

calling sequence : POPEN (NUNIT, JCONTR)

[Input] NUNIT : logical unit number (1 ~ 99),  
[Output] JCONTR : data of the Control Section (40 words).

subroutines called : GETDCB, DATE

### 3.5 PWSTAT Subroutine

PWSTAT declares the start of writing and sets the exclusive control.  
calling sequence : PWSTAT (NUNIT)

[Input] NUNIT : logical unit number  
subroutines called : none

### 3.6 PWEND Subroutine

PWEND declares the end of writing and resets the exclusive control.  
calling sequence : PWEND (NUNIT)  
  
[Input] NUNIT : logical unit number  
subroutines called : none

## 3.7 PSET Subroutine

PSET sets the node name to the Directory Section of the DATA-POOL and the address to be write. This subroutine must be called before PRITE ~ PRITE4 statements.

calling sequence : PSET (NUNIT, NODE, NTH, INFOM, NUPDAT, NRETUN)

[Input]      NUNIT : logical unit number,  
               NODE : node names from 1 to NTH levels to be set,  
               NTH : number of levels for node names to be set,  
               INFOM : user information to be set to the sub-directory  
                   section (5),  
               NUPDAT : condition flag to be update,  
                   0 = node name and address are not set when the same  
                   node and data already exist.  
                   1 = node name and address are reset although the same  
                   node already exists.

[Output]     NRETUN : return condition,  
               0 = set node name and address to be write.  
               1 = node name and address are not set because the same  
                   node exists or abnormal operations are required.

subroutines called : DCLEAR, DATDLT

The general flow of the PSET subroutine is shown in Fig. 3.1.

### 3.8 PFIND Subroutine

PFIND searches an assigned node name in the Directory Section and set the record address to be read. This subroutine must be called before PREAD ~ PREAD4 statements.

calling sequence : PFIND (NUNIT, NODE, NTH, NDIRC, LLL)

[Input]    NUNIT : logical unit number,  
            NODE : node names from 1 to NTH levels to be search,  
            NTH : number of levels for node names to be search.

[Output]    NDIRC : data of the sub-directory of the assigned node name (12),  
            LLL : return condition,  
                0 = normal return.  
                800 = sub-directory not exists.  
                1xx = node name is strange.  
                2xx = address of the directory is strange.  
                900 = level of the node name is strange.

subroutines called: none

The general flow of the PFIND subroutine is shown in Fig. 3.2.

### 3.9 PRITE ~ PRITE4 subroutines

These subroutines store the data in the Data Section of DATA-POOL. PRITE stores only comments. PRITE1 ~ PRITE4 store the one-dimensional data consisted of 1 to 4 sets, respectively.

calling sequence : PRITE (NUNIT, ICM)  
 PRITE1(NUNIT, ICM, N1, D1)  
 PRITE2(NUNIT, ICM, N1, D1, N2, D2)  
 PRITE3(NUNIT, ICM, N1, D1, N2, D2, N3, D3)  
 PRITE4(NUNIT, ICM, N1, D1, N2, D2, N3, D3, N4, D4)

[Input]    NUNIT : logical unit number,  
 ICM : comments of the data (20 words),  
 N1 ~ N4 : size of the arrays D1 ~ D4,  
 D1 ~ D4 : data arrays to be write.

subroutines called : DCLEAR, WRTCHK

### 3.10 PREAD ~ PREAD4 Subroutines

These subroutines read the data in the Data Section according to the record sequence written by PRITE ~ PRITE4 statements. PREAD reads comments and record information of the node name. PREAD1 ~ PREAD4 read the one-dimensional data of 1 to 4 sets corresponding to PRITE ~ PRITE4 statements, respectively.

calling sequence : PREAD (NUNIT, NAME1, NAME2, ICM, NASBD, NOBDS, NOARY,  
 NDATA)  
 PREAD1(NUNIT, ICM, N1, D1)  
 PREAD2(NUNIT, ICM, N1, D1, N2, D2)  
 PREAD3(NUNIT, ICM, N1, D1, N2, D2, N3, D3)  
 PREAD4(NUNIT, ICM, N1, D1, N2, D2, N3, D3, N4, D4)

[Input]    NUNIT : logical unit number,  
 NAME1 : node name of the data in the Data Section,  
 NAME2 : not used (for future use),  
 ICM : comments of the data (20 words),  
 NASBD : address for the sub-directory of the node name,  
 NOBDS: the order of the sub-data set in the Data Section,

NOARY : number of data arrays,  
NDATA : size of each array (NOARY),  
N1~N4 : size of the arrays D1~D4,  
D1~D4 : data arrays to be read.

subroutines called : none

### 3.11 PDELT Subroutine

PDELT erases directories and data under the assigned node name from the DATA-POOL.

calling sequence : PDELT (NUNIT, NODE, NTH, NRETUN)

[Input]    NUNIT : logical unit number,  
            NODE : node names to be erase,  
            NTH : number of levels for the node name.

[Output]    NRETUN: return condition,  
              0 = normal return.  
              1 = sub-directory of the node name not exist.

subroutines called : none

### 3.12 PAGET Subroutine

PAGET gets the record address assigned when the PFIND or PSET subroutine is executed. This subroutine is ordinarily used after the PFIND statements, and the obtained record address is used by the PASTO subroutine in order to read some data in the Data Section.

calling sequence : PAGET(N)

[Output]    N : direct-access record address.

subroutines called : none

### 3.13 PDGET Subroutine

PDGET gets information of the directory record assigned by the user.

calling sequence : PDGET (NUNIT, NODE, NTH, ITEM, NSDIRC)

[Input] NUNIT : logical unit number,  
NODE : node names from 1 to NTH levels,  
NTH : number of levels for the node names.  
[Output] ITEM : number of sub-directories in the directory,  
NSDIRC: information of each sub-directory  
(12, ITEM).  
subroutines called : none

### 3.14 PASTO Subroutine

PASTO sets the direct-access record address to be read or write. The PREAD/PRITE access after the PASTO statement is performed from the record address assigned by the user.

calling sequence : PASTO (N)

[Input] N : direct-access record address.  
subroutines called : none

### 3.15 PSKIP Subroutine

PSKIP skips over some records assigned by the user.  
calling sequence : PSKIP (NUNIT, N)

[Input] NUNIT : logical unit number,  
N : number of logical record to be skip.  
(An execution of PRITE statement corresponds to a  
logical record.)  
subroutines called : none

### 3.16 SETMSG Subroutine

SETMSG stores a FORTRAN error No. 232 to the variable NEMSG.  
calling sequence : SETMSG (RET, ERRNO, N1, N2)  
subroutine called : none

## 3.17 WRTCHK Subroutine

WRTCHK sets numbers of physical and logical records of data written in the DATA-POOL to the Control and the Directory Sections.

calling sequence : WRTCHK (NUNIT, IXOLD)

[Input] NUNIT : logical unit number,  
 IXOLD : direct-access record written in the Data Section.  
 subroutines called : none

## 3.18 NODEER Subroutine

NODEER is an error routine to print error message.

calling sequence : NODEER (NUNIT, NTH, NODE)

[Input] NUNIT : logical unit number,  
 NTH : number of levels for node names,  
 NODE : node names from 1 to NTH levels.

subroutines called : none

## 3.19 CATLST Subroutine

CATLST prints the Control and the Directory Sections of the DATA-POOL in order to obtain record information.

calling sequence : CATLST (NUNIT)

[Input] NUNIT : logical unit number.

## 3.20 GETDCB Subroutine

GETDCB is written in ASSEMBLER language and obtains DCB information of the DATA-POOL.

calling sequence : GETDCB (DDNAME, LRECL, LBLKS, RECFM, DSORG, IR)

[Input] DDNAME : DD name (FT91F001 etc.)[8 bytes character],  
 [Output] LRECL : record length of the direct-access data set  
               [4 bytes integer],

```
LBLKS : block size of the direct-access data
        set [4 bytes integer],
RECFM : record format of the direct-access data
        set [4 bytes character],
DSORG : data set organization [4 bytes character],
IR    : return condition,
        0 = normal return.
        8 = DD name not exists.
subroutines called : none
```

### 3.21 SUBDLT Subroutine

SUBDLT erases directories and data stored in the sub-directory.  
calling sequence : SUBDLT (NUNIT, NSDOLD)

[Input] NUNIT : logical unit number,  
NSDOLD : sub-directory to be erase (12).  
subroutines called : DIRDLT, DATDLT

### 3.22 DIRDLT Subroutine

DIRDLT erases the directory assigned in the sub-directory.  
The erase is carried out by setting '////' to the variable of node name.

calling sequence : DIRDLT (NUNIT, NSDOLD)

[Input] NUNIT : logical unit number,  
NSDOLD: sub-directory (12).  
subroutines called : none

### 3.23 DATDLT Subroutine

DATDLT erases the data assigned in the sub-directory.  
The erase is carried out by setting '////' to the variable of node name.

calling sequence : DATDLT (NUNIT, NSDOLD)

[Input]        NUNIT : logical unit number,  
                NSDOLD: sub-directory (12).  
subroutines called : none

### 3.24 DCLEAR Subroutine

DCLEAR recovers the directory when record overflow is occurred in PSET  
or PRITE subroutines.

calling sequence : DCLEAR  
subroutines called : none

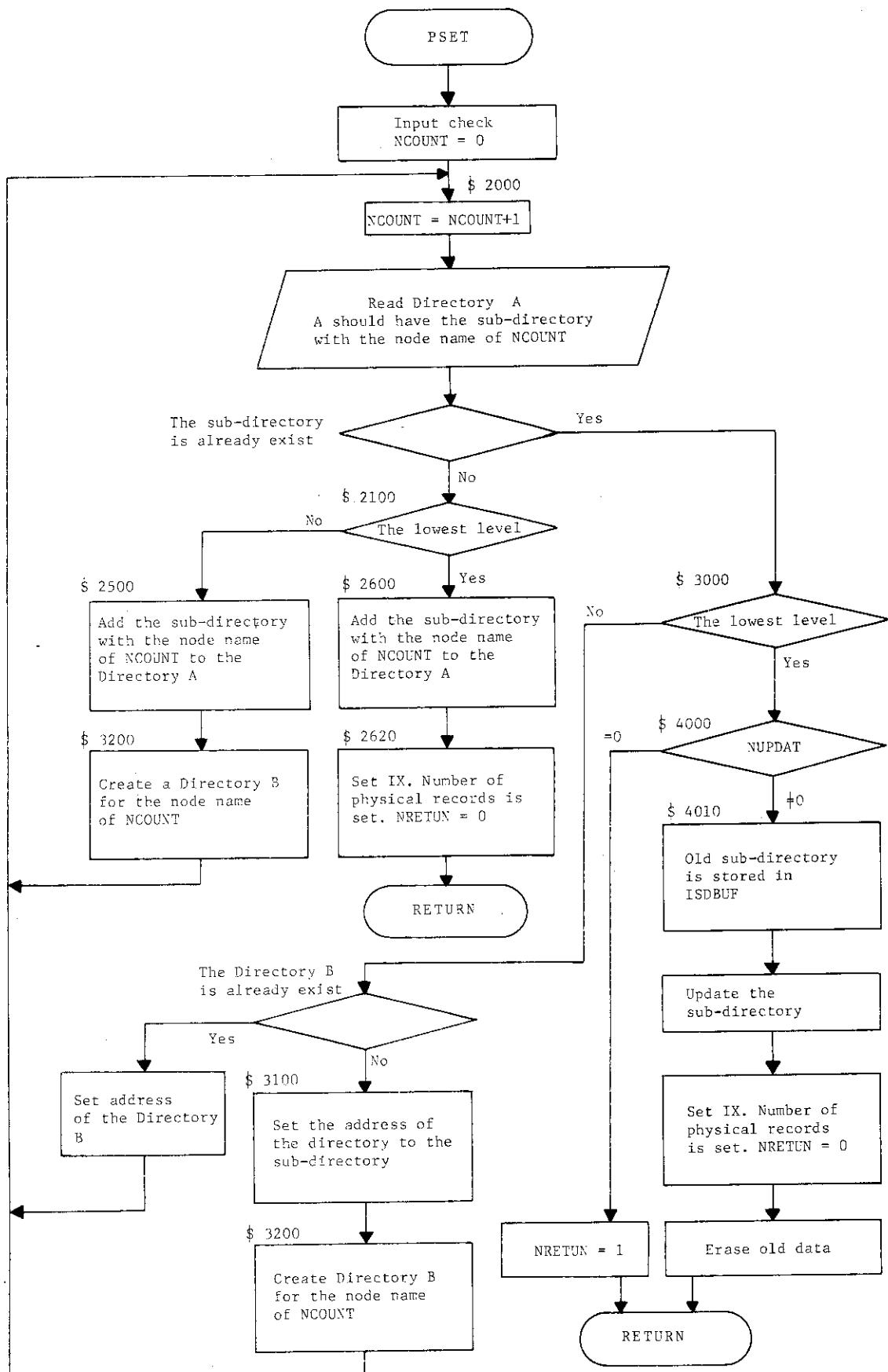


Fig. 3.1 Flow chart of PSET subroutine

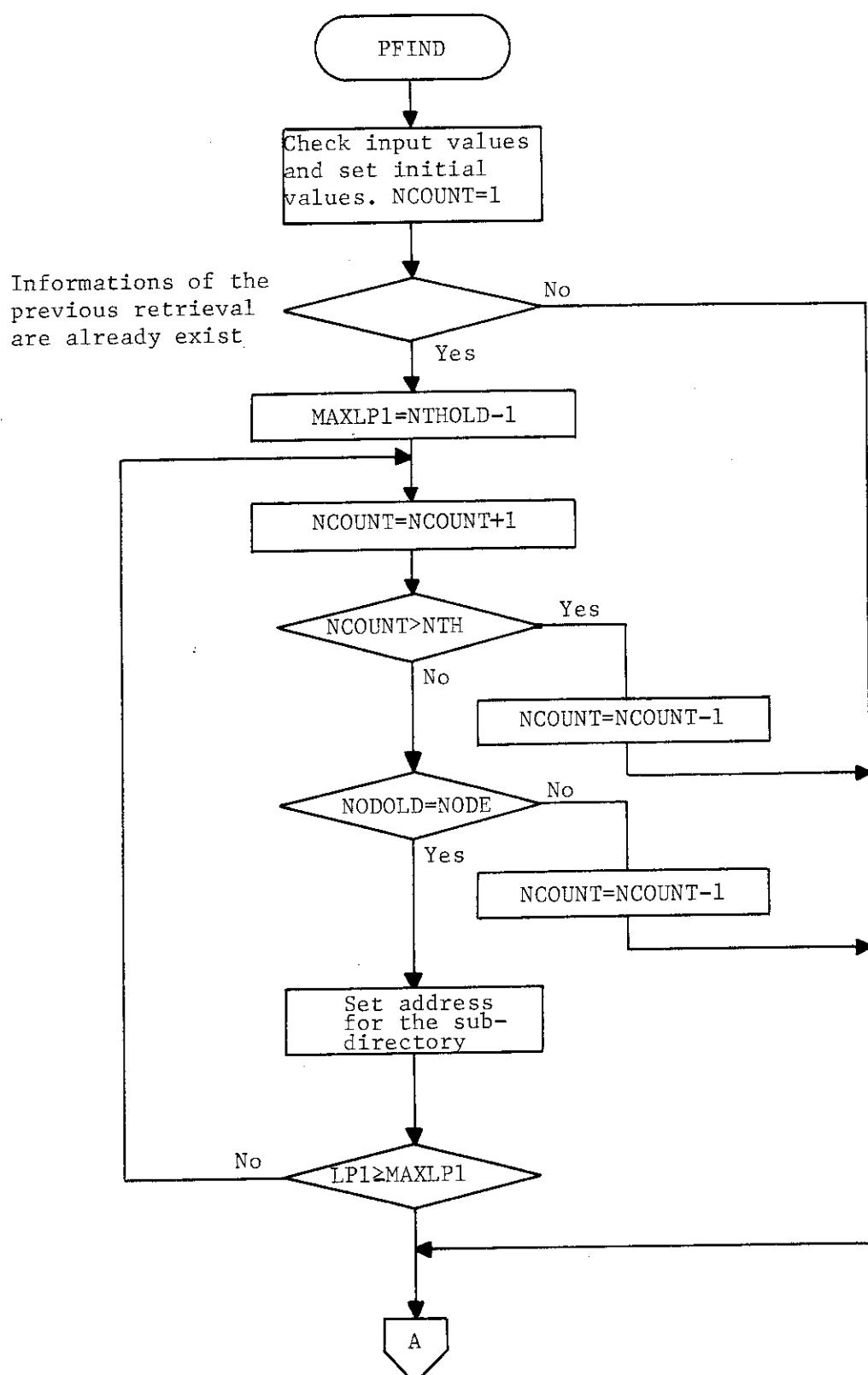


Fig. 3.2 Flow chart of PFIND subroutine

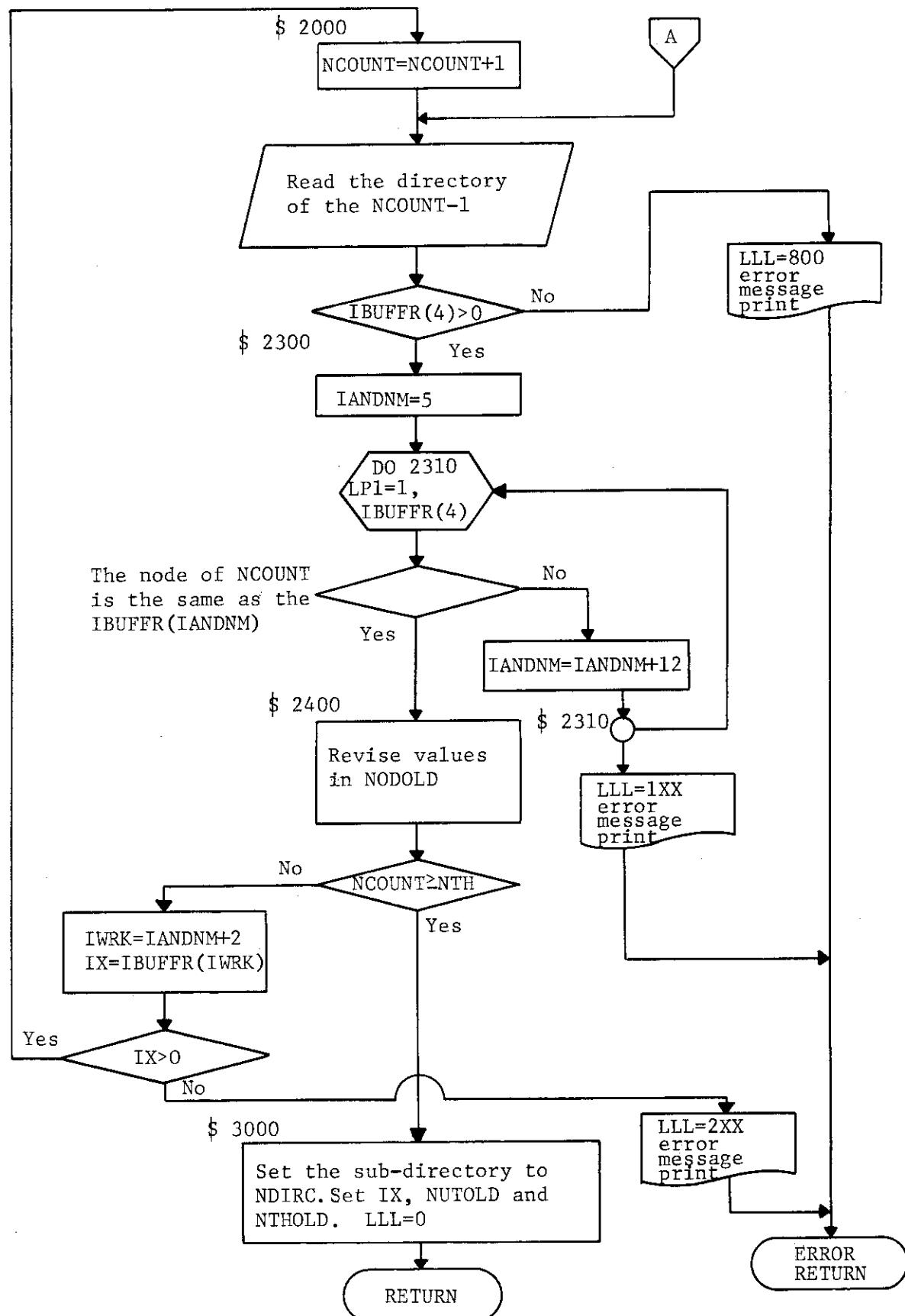


Fig. 3.2 (continued)

#### 4. Access Method for DATA-POOL

This section describes the access method of the DATA-POOL by using the access subroutines noted in the previous Section. The DATA-POOL is a direct-access data set, so that an initialization must be performed when the data set is allocated. The initialization is carried out by using INIT command in a TSS Management utility POOL described in the next Section. The allocation of the data set with DD statement is as follows :

```
//FT01F001 DD DSN=J3679.DATAPOOL.DATA, UNIT=D0954,
// SPACE=(TRK,(50,10)), DCB=(LRECL=3600, BLKSIZE=3600, RECFM=F),
// LABEL=(,,, OUT)
```

or

```
// EXPAND DISKTO, DDN=FT01F001, DSN=J3679.DATAPOOL.DATA,
```

```
// MODE=OUT
```

or on TSS terminal

```
ATTR DCB LR(3600) BL(3600) REC(F)
```

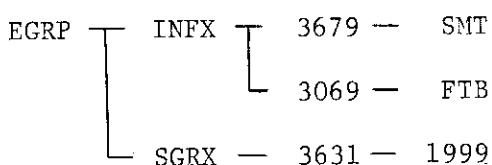
```
ALLOC DA (DATAPOOL.DATA) UNIT(D0954) SP(50 10) T US(DCB) CAT
```

The initialized data set can be used as a DATA-POOL. A POPEN statement must be called in the user's program in order to access the DATA-POOL before the other access statements appear.

##### 4.1 Generation of Node Structure

The sequence generating node structure is described by using a sample as follows :

A sample node structure is



The generation of the node structure is carried out by using PSET, PWSTAT, PWEND and POPEN statements.

```
CHARACTER JCONTR(40), NODE(10), INFOM(5)
CALL POPEN (1, JCONTR)
CALL PWSTAT(1)
```

```

NTH=1
NODE(1)='EGRP'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH 'EGRP' CAN BE WRITTEN AT THE POSITION

NTH=2
NODE(2)='INFX'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH 'INFX' CAN BE WRITTEN AT THE POSITION

NTH=3
NODE(3)='3679'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH '3679' CAN BE WRITTEN AT THE POSITION

NTH=3
NODE(3)='3069'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH '3069' CAN BE WRITTEN AT THE POSITION

NTH=3
NODE(3)='3679'
NTH=4
NODE(4)=' SMT'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH ' SMT' CAN BE WRITTEN AT THE POSITION

NODE(3)='3069'
NODE(4)=' FTB'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH ' FTB' CAN BE WRITTEN AT THE POSITION

NODE(2)='SGRX'
NTH=2
CALL PSET(1, NODE, NTH, INFOR, O, L)
NODE(3)='3631'
NTH=3
CALL PSET(1, NODE, NTH, INFORM, O, L)
C---DATA WITH '3631' CAN BE WRITTEN AT THE POSITION

NTH=4
NODE(4)='1999'
CALL PSET(1, NODE, NTH, INFOM, O, L)
C---DATA WITH '1999' CAN BE WRITTEN AT THE POSITION

CALL PWEND(1)

```

If a node name assigned by the user already exist in the DATA-POOL, the variable L in the PSET subroutine is set to 1, and the registration of the node name is not executed by the condition of NUPDAT=0. In the period executing between PWSTAT and PWEND statements, the other write accesses to the DATA-POOL is inhibited. The user information of 5 words for each node

can be stored by using the PSET statement to set the variables of INFOM. Arbitrary tree structures can be generated by the user as shown in the sample problem.

#### 4.2 Storage Procedure

PRITE, PRITE1, PRITE2, PRITE3 and PRITE4 statements are prepared to store data in the Data Section of the DATA-POOL.

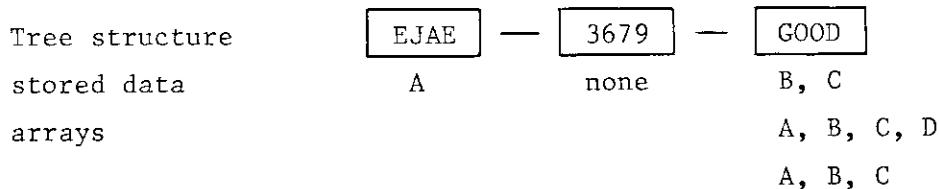
These statements must be located after the PSET statement. The combination of PRITE statements is arbitrarily defined by the user. A sample is shown as follows :

```

CHARACTER JCONTR(40), NODE(10), ICM(20)
DIMENSION A(100), B(200), C(300), D(400), INFOM(5)
IA=100
IB=200
IC=300
ID=400
CALL POPEN(1, JCONTR)
CALL PWSTAT(1)
NTH=1
NODE(1)='EJAE'
CALL PSET(1, NODE, NTH, INFOM, 0, L)
IF(L.NE.0) GO TO 7
CALL PRITE1(1, ICM, IA, A)
7   NTH=2
NODE(2)='3679'
CALL PSET(1, NODE, NTH, INFOM, 0, L)
IF(L.NE.0) GO TO 77
CALL PRITE(1, ICM)
77  NTH=3
NODE(3)='GOOD'
CALL PSET(1, NODE, NTH, INFOM, 0, L)
IF(L.NE.0) GO TO 777
CALL PRITE2(1, ICM, IB, B, IC, C)
CALL PRITE4(1, ICM, IA, A, IB, B, IC, C, ID, D)
CALL PRITE3(1, ICM, IA, A, IB, B, IC, C)
777 CALL PWEND(1)

```

The above sample generates the following tree structure and stores data of arrays A, B, C and D.



The comments for data can be stored by setting the variable ICM before PRITE ~ PRITE4 statements. Many kinds of data can be stored with a node name. The record sequence can be arbitrarily defined by the user.

#### 4.3 Retrieval Procedure

Node names and data stored in the DATA-POOL are retrieved by using PFIND, PREAD, PREAD1, PREAD2, PREAD3 and PREAD4 statements. PDGET, PAGET, PASTO and PSKIP statements may be also used to retrieve the data skillfully. A sample procedure is shown for the data stored in the previous Section.

```

CHARACTER JCONTR(40), NODE(10), ICM(20)
DIMENSION A(100), B(200), C(300), D(400), INFOM(5), NDIRC(12)
DIMENSION NDATA(4)

IA=0
IB=0
IC=0
ID=0
NTH=1
NODE(1)='EJAE'
CALL PFIND(1, NODE, NTH, NDIRC, L)
IF(L.NE.0) GO TO 999
INFOM(1)=NDIRC(8)
INFOM(2)=NDIRC(9)
INFOM(3)=NDIRC(10)
INFOM(4)=NDIRC(11)
INFOM(5)=NDIRC(12)
CALL PREAD1(1, ICM, IA, A)
NTH=2
NODE(2)='3679'

```

```

CALL PFIND(1, NODE, NTH, NDIRC, L)
IF(L.NE.0) GO TO 999
DO 10 I=1, 5
K=I + 7
10 INFOM(I)=NDIRC(K)
CALL PREAD(1, N1, N2, ICM, N3, N4, N5, NDATA)
NTH=3
NODE(3)='GOOD'
CALL PFIND(1, NODE, NTH, NDIRC, L)
IF(L.NE.0) GO TO 999
DO 11 I=1, 5
K=I + 7
11 INFOM(I)=NDIRC(K)
CALL PREAD2(1, ICM, IB, B, IC, C)
CALL PREAD4(1, ICM, IA, A, IB, B, IC, C, ID, D)
CALL PREAD3(1, ICM, IA, A, IB, B, IC, C)
:
C---ERROR MESSAGE DISPLAY
999 CALL NODEER(1, NTH, NODE)

```

In the sequence of the sample procedure, the variables IA, IB, IC and ID are initially set zero. However, the variables are set the sizes of data arrays after PREAD statements, so that the initial setting may not be necessary. Information defined by the user are stored in NDIRC(8) ~ NDIRC(12) arrays after PREAD statements. An error monitor will be located at the end of the user's program in order to detect errors caused by mistaken conditions.

If the user wish to read only some parts of data, PSKIP statement may be used as follows :

```

NTH=3
NODE(3)='GOOD'
CALL PFIND(1, NODE, NTH, NDIRC, L)
IF(L.NE.0) GO TO 999
CALL PREAD2(1, ICM, IB, B, IC, C)
CALL PSKIP(1, 1)
CALL PREAD3(1, ICM, IA, A, IB, B, IC, C)
:
:
```

PAGET and PASTO statements may be used to read data in a same category for each node name as follows :

```
DO 100 N=1, 10
CALL PSET(1, NODE(1, N), NTH, INFOM, 0, L)
DO 100 M=1, 20
100   CALL PRITE1(1, ICM, N1, A(1, M))
      :
      :
      :
      :
DO 200 N=1, 10
CALL PFIND(1, NODE(1, N), NTH, NDIRC, L)
200   CALL PAGET(NADRS(N))
      :
      :
      :
      :
DO 300 M=1, 20
DO 300 N=1, 10
CALL PASTO(NADRS(N))
CALL PREAD1(1, ICM, N1, B)
CALL PAGET (NADRS(N))
      :
      :
      :
      :
300   CONTINUE
```

A sample program to retrieve information of nuclei and atomic number densities for each material in the macroscopic cross-section library is shown in Fig. 4.1. A Job Control Language and the input data are also shown in this figure.

## 4.4 Limitations and Notes for Operation

## a) Limitations

- (i) Logical unit number is allowed from FT01F001 to FT99F001.
- (ii) The maximum level for node names is 10.
- (iii) Physical record length in the DATA-POOL is allowed up to 1000 words.<sup>+</sup>
- (iv) DATA-POOL is a direct-access data base so that a direct-access device is essentially required.
- (v) Access subroutines for DATA-POOL are written in FORTRAN77 language so that the FORTRAN77 compiler is essentially needed.

## b) Notes for Operation

- (i) Initialization of the DATA-POOL is executed for the area allocated with DD statement. The number of initialized records is printed in PINIT subroutine, so that user should not access over the limit.
- (ii) In the period executing between PWSTAT and PWEND statements, the other write access is inhibited. Namely the other jobs may be terminated.
- (iii) When the job is abnormally terminated, the DATA-POOL may not be generated correctly. The restoration can be performed by using following procedures.
  - write flag is already on. → execute FLAG command of the TSS management utility POOL.
  - node name is registered, → execute DELETE command of the TSS management utility POOL.
  - writing data exceeds the → execute MEND command of the TSS limit of the initialized management utility POOL.
  - records or the other destructions.

The description of the TSS management utility POOL is shown in the next Section.

---

+ : The value can be changed by modifications of common blocks, PINIT and POPEN subroutines.

- (iv) The node name consists of 4-characters. The characters A~Z, 0~9 and blank are allowed. For example, node names \_SMT and SMT\_ are different from each other.

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 0001

C BLOCKD LEVEL=1 DATE=84.03.14 00000100  
 BLOCK DATA 00000200  
 COMMON/B/MAT(141) 00000300  
 COMMON/C/MCR(3,141) 00000400  
 CHARACTER\*4 MCR 00000500  
 DATA MAT/ 1128,1129,1130,1131,1169,1195,1270,1031,1032,1033,1120,00000600  
 1 1146,1170,1171,1172,1173,1174,1175,1176,1177,1178,1181,1182,1183,00000700  
 2 1184,1185,1186,1196,1027,1030,1083,1084,1125,1127,1137,1138,1139,00000800  
 3 1141,1149,1150,1156,1160,1043,1050,1056,1057,1161,1162,1163,1269,00000900  
 4 1271,1272,1273,1289,1294,1296,1193,1194,1280,1190,1191,1192,1261,00001000  
 5 1264,1265,1266,1297,1274,1275,1276,1288,1197,1260,1262,1263,1286,00001100  
 6 1287,1199,1290,1291,1292,1293,1295,1189,1277,1281,1282,1283,1284,00001200  
 7 1285,6156,6193,6197,6199,6261,6262,6263,6264,6271,6273,6283,6296,00001300  
 8 6406,6407,6410,6411,6412,6414,6415,6416,6417,6418,6419,6420,6421,00001400  
 9 6422,6423,2600,2721,2722,2723,2724,2725,2726,7824, 0, 0, 0,00001500  
 A 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0/00001600  
 DATA ((MCR(I,J),I=1,3),J=1,58)/'74- ','W-18','2 ','74- ','W-18',00001700  
 1 '3 ','74- ','W-18','4 ','74- ','W-18','6 ','1- ','H- ', 00001800  
 2 '3 ','20-C','A ',' ','2-H','E- ','4 ','66-D','Y-16', 00001900  
 3 '4 ','71-L','U-17','5 ','71-L','U-17','6 ','1- ','H- ', 00002000  
 4 '2 ','2-H','E- ','3 ','54-X','E-12','4 ','54-X','E-12', 00002100  
 5 '6 ','54-X','E-12','8 ','54-X','E-12','9 ','54-X','E-13', 00002200  
 6 '0 ','54-X','E-13','1 ','54-X','E-13','2 ','54-X','E-13', 00002300  
 7 '4 ','54-X','E-13','6 ','36-K','R- 7','8 ','36-K','R- 8', 00002400  
 8 '0 ','36-K','R- 8','2 ','36-K','R- 8','3 ','36-K','R- 8', 00002500  
 9 '4 ','36-K','R- 8','6 ','23- ','V ',' ','62-S','M-14', 00002600  
 A '9 ','64-G','D ',' ','75-R','E-18','5 ','75-R','E-18', 00002700  
 B '7 ','45-R','H-10','3 ','73-T','A-18','2 ','43-T','C- 9', 00002800  
 C '9 ','47-A','G-10','7 ','47-A','G-10','9 ','55-C','S-13', 00002900  
 D '3 ','17-C','L ',' ','19- ','K ',' ','11-N','A- 2', 00003000  
 E '3 ','5- ','B- 1','1 ','92- ','U-23','4 ','94-P','U-23', 00003100  
 F '8 ','95-A','M-24','1 ','95-A','M-24','3 ','94-P','U-24', 00003200  
 G '2 ','96-C','M-24','4 ','92- ','U-23','6 ','1- ','H- ', 00003300  
 H '1 ','3-L','I- ','6 ','3-L','I- ','7 ','5- ','B- 1', 00003400  
 I '0 ','4-B','E- ','9 ','54-X','E-13','5 ','90-T','H-23', 00003500  
 J '2 ','13-A','L- 2','7 ','14-S','I ',' ',' ', 00003600  
 DATA ((MCR(I,J),I=1,3),J=59,116)/ '12-M','G ',' ','28-N', 00003700  
 1 'I ',' ','24-C','R ',' ','26-F','E ',' ','92- ', 00003800  
 2 'U-23','5 ','94-P','U-23','9 ','94-P','U-24','0 ','94-P', 00003900  
 3 'U-24','1 ','93-P','U-23','3 ','6- ','C- 1','2 ','7- ', 00004000  
 4 'N- 1','4 ','8- ','0- 1','6 ','82-P','B ',' ','25-M', 00004100  
 5 'N- 5','5 ','92- ','U-23','3 ','92- ','U-23','8 ','93-N', 00004200  
 6 'P-23','7 ','22-T','I ',' ','42-M','O ',' ','27-C', 00004300  
 7 '0- 5','9 ','63-E','U-15','1 ','63-E','U-15','3 ','63-E', 00004400  
 8 'U-15','2 ','63-E','U-15','4 ','29-C','U ',' ','41-N', 00004500  
 9 'B- 9','3 ','9- ','F ',' ','48-C','D ',' ','48-C', 00004600  
 A 'D-11','3 ','79-A','U-19','7 ','40-Z','IRC- ','2 ','73-T', 00004700  
 B 'A-18','1 ','11-N','A- 2','3 ','13-A','L- 2','7 ','25-M', 00004800  
 C 'N- 5','5 ','27-C','0- 5','9 ','92- ','U-23','5 ','92- ', 00004900  
 D 'U-23','8 ','93-N','P-23','7 ','92-P','U-23','9 ','3-L', 00005000

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 0001

Fig. 4.1 Sample program for information retrieval of DATA-POOL

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

```

SUBROUTINE NUMBRP(X,Y,IH,A,M,L,IC)
CHARACTER KL*1
CHARACTER DL*2
CHARACTER ZL*4
IDEC=0
ICHK=0
B=A
IF(B.EQ.0.0) GO TO 100
IF(B.GT.0.0) GO TO 20
KL='-
CALL GSCHAR(X,Y,IH,KL,IC,1)
B=ABS(B)
ICHK=1
20 ADEC=ALOG10(B)
IF(ADEC.LT.0.) ADEC=ADEC-1
IDEC=ADEC
JDEC=IABS(IDEC)
IF(IDEC.EQ.0) GO TO 10
IF(IDEC.GT.0) GO TO 2
AA=B*10.**JDEC
GO TO 3
2 AA=B*10./(10.**JDEC)
GO TO 3
10 AA=B
3 CONTINUE
CC      SHI-SHA GO-NYU
ABT=1.0
IF(M.GT.0) ABT=10.**M
ABL=AA*ABT
IABL=IFIX(ABL)
ABL=ABL-FLOAT(IABL)
IF(ABL.GE.0.5) IABL=IABL+1
JBL=IABL/IFIX(ABT)
WRITE(KL,1) JBL
00000100
00000200
00000300
00000400
00000500
00000600
00000700
00000800
00000900
00001000
00001100
00001200
00001300
00001400
00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200
00002300
00002400
00002500
00002600
00002700
00002800
00002900
00003000
00003100
00003200
00003300
00003400

```

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 0001

```

1 FORMAT(I1)          00003500
XX=999.              00003600
YY=999.              00003700
IF(ICHK.EQ.0) XX=X   00003800
IF(ICHK.EQ.0) YY=Y   00003900
CALL GSCHAR(XX,YY,IH,KL,IC,1) 00004000
IF(M.EQ.0) GO TO 5   00004100
KL='.'               00004200
CALL GSCHAR(999.,999.,IH,KL,IC,1) 00004300
DO 4 I=1,M           00004400
IABL=10*(IABL-JBL*IFIX(ABT)) 00004500
JBL=IABL/IFIX(ABT)       00004600
WRITE(KL,1) JBL       00004700
CALL GSCHAR(999.,999.,IH,KL,IC,1) 00004800
4 CONTINUE            00004900
5 CONTINUE            00005000
IF(IDEC.EQ.0) GO TO 1000 00005100
IF(L.EQ.0) GO TO 6    00005200
KL='*'               00005300
CALL GSCHR1(999.,999.,IH,KL,IC,1,9) 00005400
DL='10'               00005500
CALL GSCHAR(999.,999.,IH,DL,IC,2) 00005600
KL='-'               00005700
JH=IH-1              00005800
YYY=Y+2.7             00005900
IF(IDECLT.0) CALL GSCHAR(999.,YYY,JH,KL,IC,1) 00006000
IF(JDEC.LT.10) GO TO 8 00006100
WRITE(DL,7) JDEC      00006200
00006300
7 FORMAT(I2)          00006400
CALL GSCHAR(999.,YYY,JH,DL,IC,2) 00006500
GO TO 1000            00006600
8 WRITE(KL,1) JDEC     00006700
CALL GSCHAR(999.,YYY,JH,KL,IC,1) 00006800
GO TO 1000            00006900
6 CONTINUE            00007000
KL='E'               00007100
CALL GSCHAR(999.,999.,IH,KL,IC,1) 00007200
KL='-'               00007300
IF(IDECLT.0) CALL GSCHAR(999.,999.,IH,KL,IC,1) 00007400
IF(JDEC.LT.10) GO TO 9 00007500
WRITE(DL,7) JDEC      00007600
CALL GSCHAR(999.,999.,IH,DL,IC,2) 00007700
GO TO 1000            00007800
9 WRITE(KL,1) JDEC     00007900
CALL GSCHAR(999.,999.,IH,KL,IC,1) 00008000
GO TO 1000            00008100
100 ZL='0.0'            00008200
CALL GSCHAR(X,Y,IH,ZL,IC,4) 00008300
1000 CONTINUE          00008400
RETURN               00008500
E N D

```

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 0001

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

```

C      PRGRP.          LEVEL=3          DATE=83.08.03          00000010
C      SUBROUTINE PRGRP(ING,GNG,IGG,GGG,I06)          00000020
C
C      PRGRP PRINTS THE NEUTRON - GAMMA-RAY ENERGY GROUP STRUCTURE. 00000030
C      MAXIMUM NUMBER OF NEUTRON GROUP IS 200 AND GAMMA GROUP IS 50. 00000040
C
C      DIMENSION GNG(1),GGG(1)          00000050
C      NG=MAX0(ING,IGG)          00000060
C      IF(NG.GT.1000) RETURN          00000070
C      DO 50 J=1,NG,100          00000080
C      WRITE(I06,950)          00000090
C      IPOS=J          00000100
C      IEND=J+49          00000110
C      IF(CIEND.GT.NG)IEND=NG          00000120
C      DO 50 I=IPOS,IEND          00000130
C      I50=I+50          00000140
C      IF(I.GT.ING .OR. I.GT.IGG) GO TO 30          00000150
C      IF(I50.GT.ING) GO TO 20          00000160
C      WRITE(I06,1000) I,GNG(I),GNG(I+1),I50,GNG(I50),GNG(I50+1), 00000170
C      *          I,GGG(I),GGG(I+1)          00000180
C      *          GO TO 50          00000190
C      20 WRITE(I06,1100) I,GNG(I),GNG(I+1),I,GGG(I),GGG(I+1)          00000200
C      *          GO TO 50          00000210
C      30 IF(I50.GT.ING) GO TO 35          00000220
C      *          WRITE(I06,1000) I,GNG(I),GNG(I+1),I50,GNG(I50),GNG(I50+1) 00000230
C      *          GO TO 50          00000240
C      35 IF(I.GT.ING) GO TO 40          00000250
C      *          WRITE(I06,1000) I,GNG(I),GNG(I+1)          00000260
C      *          GO TO 50          00000270
C      40 WRITE(I06,1200) I,GGG(I),GGG(I+1)          00000280
C      50 CONTINUE          00000290
C      950 FORMAT('1'//40X,'ENERGY GROUP STRUCTURE'/          00000300
C      *          '0',25X,'--- NEUTRON GROUP ---',35X,'--- GAMMA GROUP          00000310
C      *          '----', '0', 5X,'GROUP          ENERGY RANGE',11X,'GROUP          ENERGY          00000320
C      *          'RANGE', 16X,'GROUP          ENERGY RANGE')          00000330
C      1000 FORMAT(5X,I4,1P2E13.4,5X,I4,2E13.4,10X,I4,2E13.4)          00000340
C      1100 FORMAT(5X,I4,1P2E13.4,45X,I4,2E13.4)          00000350
C      1200 FORMAT(80X,I4,1P2E13.4)          00000360
C      RETURN          00000370
C      END          00000380
C
C
C      TABLE PRODUCTION OF MATERIALS CONTAINED IN JSD1000 LIBRARY          00000390
C      LEVEL 2.0      +REVISED JULY 29 1983          00000400
C      LEVEL 3.0      +REVISED AUG. 3 1983          00000500
C      FORTRAN77 LEVEL 1.0      +REVISED MAR. 14 1984

```

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000:

```

COMMON/A/MID(30,5),AD(30,5),TP(30),MFID(30),NM(5),ND(12)      00000600
COMMON/B/MAT(141)                                              00000700
COMMON/C/MCR(3,141)                                             00000800
DIMENSION EG(51),AG(101),MPC(30),JCONTR(40),NSDIRC(12,74)     00000900
DIMENSION D(1024),MD1(30),AD1(30)                                00001000
DIMENSION LID(10,74),MSDIRC(12,74),EN(4000),DD(1000)           00001100
COMMON/DPCONT/LCONTR,NCONTR,ICONTR(40,3),IX,NSUBDS            00001200
COMMON/DPWORK/LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00001300
+                      NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,NTHOLD, 00001400
+                      NODOLD(10,2),NA1                           00001500
+                           00001600
C
CHARACTER*4 MCR                                              00001700
CHARACTER*4 NODE(3),MGB(3),ITCM(20,74),ICM(20),MMID(30)        00001800
CHARACTER*4 NULTX,NSELF,NSGRX,NINFX,NFX,NIDN,NTEMPA,NTEMPB,MCHR 00001900
CALL PLOTS(D,1024)                                            00002000
CALL NEWPEN(2)                                                 00002100
1 READ(5,2,END=1000) IN,NODE(1),NODE(2),ICONT1,ICONT2          00002200
2 FORMAT(15,2A4,2I5)                                           00002300
CALL POPEN(IN,JCONTR)                                         00002400
C   NODE(1)='EGRP'                                              00002500
C   NODE(2)='FX32'                                              00002600
C   NODE(2)='INFX'                                              00002700
C   NODE(2)='SGRX'                                              00002800
NULTX='ULTX'                                                 00002900
NSELF='SELF'                                                 00003000
NSGRX='SGRX'                                                 00003100
NINFX='INFX'                                                 00003200
NFX='FX'                                                    00003300
CALL PFIND(IN,NODE,1,ND,LLL)                                 00003400
IF(LL.LEQ.0) GO TO 4                                         00003500
WRITE(6,3) LLL,IN,NODE(1)                                     00003600
3 FORMAT(5X,'PFIND ERROR CODE',I5,' IN UNIT',I3,' OF NODE ',A4) 00003700
GO TO 1000                                                 00003800
4 ING=ND(8)                                                 00003900
IGG=ND(9)                                                 00004000
IF(ING.EQ.0) GO TO 8800                                      00004100
IF(IGG.EQ.0) GO TO 8810                                      00004200
CALL PREAD2(IN,ICM,ING+1,EN,IGG+1,EG)                      00004300
GO TO 8890                                                 00004400
8810 CALL PREAD1(IN,ICM,ING+1,EN)                            00004500
GO TO 8890                                                 00004600
8800 CALL PREAD1(IN,ICM,IGG+1,EG)                            00004700
8890 CALL PRGRP(ING,EN,IGG,EG,6)                            00004800
C
IF(NODE(1).EQ.NULTX) GO TO 776                             00004900
8891 FORMAT(A2,2X)                                           00005000
READ(NODE(2),8891) NIDN                                    00005100
CALL PFIND(IN,NODE,2,ND,LLL)                                 00005200
IF(LL.LEQ.0) GO TO 5                                         00005300
WRITE(6,3) LLL,IN,NODE(2)                                     00005400
GO TO 1000                                                 00005500
5 IPO=ND(8)                                                 00005600
IF(NIDN.NE.NFX) GO TO 776                                  00005700
CALL PREAD1(IN,ICM,IPO+1,AG)                               00005800
WRITE(6,7) IPO                                              00005900
                                         00006000

```

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000:

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

```

7 FORMAT(1H1,///,8X,I3,' ANGULAR MESH POINTS ARE DEFINED',/)      00006100
DO 25 J=1,IP0+1          00006200
25 WRITE(6,26) J,AG(J)    00006300
26 FORMAT(10X,I4,4X,1PE12.5) 00006400
C
776 WRITE(6,777)          00006500
777 FORMAT(1H1)
IF(NODE(1).NE.NULTX) CALL PDGET(IN,NODE,2,IMAX,NSDIRC) 00006600
IF(NODE(1).EQ.NULTX) CALL PDGET(IN,NODE,1,IMAX,NSDIRC) 00006700
IF(NIDN.NE.NFX) GO TO 6000 00006800
NMAX=IMAX/5+1          00006900
JJ=0                    00007000
MATMAX=141              00007100
C
C
DO 10 M=1,NMAX          00007200
C
DO 11 MM=1,5            00007300
DO 11 KK=1,30           00007400
MID(KK,MM)=0             00007500
AD(KK,MM)=0.0            00007600
TP(KK)=0.0               00007700
MFID(KK)=0               00007800
MPC(KK)=0                00007900
00008000
00008100
00008200
00008300
00008400
00008500
00008600
00008700
00008800
00008900
00009000
00009100
00009200
00009300
00009400
00009500
00009600
00009700
00009800
00009900
00010000
00010100
00010200
00010300
00010400
00010500
00010600
00010700
00010800
00010900
00011000
00011100
00011200
00011300
00011400
00011500
11 CONTINUE
DO 12 K=1,5
IF(JJ.GE.IMAX) GO TO 200
JJ=JJ+1
WRITE(NODE(3),6) NSDIRC(1,JJ)
CALL PFIND(IN,NODE,3,ND,LLL)
IF(LLL.EQ.0) GO TO 13
WRITE(6,3) LLL,IN,NODE(3)
GO TO 1000
13 MATID=ND(8)
IHS=ND(9)
IHT=ND(10)
IHM=ND(11)
CALL PREAD4(IN,ICM,NMMMM,MD1(1),NMA,MFID(1),NMAT,AD1(1),
+                               NMAT,TP(1))
NM(K)=NMMMM
NMTT=NM(K)
DO 501 L=1,20
READ(ICM(L),6) ITCM(L,K)
501 CONTINUE
DO 500 MJM=1,NMTT
MID(MJM,K)=MD1(MJM)
500 AD(MJM,K)=AD1(MJM)
12 CONTINUE
K=6
200 K=K-1
IF(K.LE.0) GO TO 10
DO 5555 ITEST=1,K
IITEST=NM(ITEST)
WRITE(6,800) ITEST,IITEST,(MID(NNNN,ITEST),NNNN=1,IITEST)
5555 WRITE(6,801) (AD(NNNN,ITEST),NNNN=1,IITEST)

```

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 000

```

800 FORMAT(5X,'K=',I3,'NMAX=',I3,' MID=',10I6)          00011600
801 FORMAT(15X,'AD=',1P10E10.3)                      00011700
      MTMAX=NM(1)                                     00011800
      DO 14 I=1,K                                     00011900
      NMT=NM(I)                                      00012000
      DO 15 J=1,NMT                                 00012100
      IF(I.NE.1) GO TO 16                            00012200
      MPC(J)=MID(J,I)
      GO TO 15                                       00012300
16   CONTINUE                                         00012400
      DO 17 JM=1,MTMAX                           00012500
      IF(MPC(JM).EQ.MID(J,I)) GO TO 15            00012600
17   CONTINUE                                         00012700
      00012800
      00012900
C     MTMAX=MTMAX+1                                00013000
      MPC(MTMAX)=MID(J,I)                         00013100
15   CONTINUE                                         00013200
14   CONTINUE                                         00013300
C     00013400
C     MAKING TABLES                               00013500
C     00013600
      CALL GSCHARC(50.,212.,3,JCONTR(1),211,80)    00013700
      CALL GSCHARC(41.,203.,3,' MATERIAL NAME',211,14) 00013800
      CALL GSCHARC(47.,193.,3,' NODE NAME',211,10)    00013900
      IF(ICONT1.NE.0) GO TO 700                     00014000
      CALL GSCHARC(41.,183.,3,' NUCLIDE NUMBER',211,15) 00014100
      GO TO 701                                     00014200
700   CALL GSCHARC(47.,183.,3,' NUCLIDE',211,9)    00014300
701   CONTINUE                                         00014400
      DO 18 JM=1,MTMAX                           00014500
      YY=172.-(JM-1)*7.
      DO 188 LKJ=1,MATMAX                         00014600
      IF(MPC(JM).EQ.MAT(LKJ)) GO TO 189           00014700
188   CONTINUE                                         00014800
      LKJ=MATMAX                                    00014900
189   CONTINUE                                         00015000
      00015100
      WRITE(6,600) JM,MPC(JM),(MCR(IKJ,LKJ),IKJ=1,3) 00015200
600   FORMAT(5X,'MPC(',I3,',')=',I6,2X,3A4)        00015300
      IF(ICONT1.NE.0) GO TO 1234                  00015400
      READ(MPC(JM),20) MCHR
      CALL GSCHARC(53.,YY,3,MCHR,211,4)
      GO TO 4321                                   00015500
      00015600
      00015700
      00015800
      00015900
      00016000
      00016100
      00016200
      00016300
      00016400
      00016500
      00016600
      00016700
      00016800
      00016900
      00017000
1234  MGB(1)=MCR(1,LKJ)
      MGB(2)=MCR(2,LKJ)
      MGB(3)=MCR(3,LKJ)
      IF(LKJ.NE.MATMAX) GO TO 4320
      MGB(1)='MAT('
      WRITE(MGB(2),20) MPC(JM)
      MGB(3)=')
4320  CONTINUE                                         00016400
      CALL GSCHARC(48.,YY,3,MGB,211,12)            00016500
4321  CONTINUE                                         00016600
      00016700
      00016800
      00016900
      00017000
      00017100
      00017200
      00017300
      00017400
      00017500
      00017600
      00017700
      00017800
      00017900
      00018000
      00018100
      00018200
      00018300
      00018400
      00018500
      00018600
      00018700
      00018800
      00018900
      00019000
      00019100
      00019200
      00019300
      00019400
      00019500
      00019600
      00019700
      00019800
      00019900
      00020000
      00020100
      00020200
      00020300
      00020400
      00020500
      00020600
      00020700
      00020800
      00020900
      00021000
      00021100
      00021200
      00021300
      00021400
      00021500
      00021600
      00021700
      00021800
      00021900
      00022000
      00022100
      00022200
      00022300
      00022400
      00022500
      00022600
      00022700
      00022800
      00022900
      00023000
      00023100
      00023200
      00023300
      00023400
      00023500
      00023600
      00023700
      00023800
      00023900
      00024000
      00024100
      00024200
      00024300
      00024400
      00024500
      00024600
      00024700
      00024800
      00024900
      00025000
      00025100
      00025200
      00025300
      00025400
      00025500
      00025600
      00025700
      00025800
      00025900
      00026000
      00026100
      00026200
      00026300
      00026400
      00026500
      00026600
      00026700
      00026800
      00026900
      00027000
      00027100
      00027200
      00027300
      00027400
      00027500
      00027600
      00027700
      00027800
      00027900
      00028000
      00028100
      00028200
      00028300
      00028400
      00028500
      00028600
      00028700
      00028800
      00028900
      00029000
      00029100
      00029200
      00029300
      00029400
      00029500
      00029600
      00029700
      00029800
      00029900
      00030000
      00030100
      00030200
      00030300
      00030400
      00030500
      00030600
      00030700
      00030800
      00030900
      00031000
      00031100
      00031200
      00031300
      00031400
      00031500
      00031600
      00031700
      00031800
      00031900
      00032000
      00032100
      00032200
      00032300
      00032400
      00032500
      00032600
      00032700
      00032800
      00032900
      00033000
      00033100
      00033200
      00033300
      00033400
      00033500
      00033600
      00033700
      00033800
      00033900
      00034000
      00034100
      00034200
      00034300
      00034400
      00034500
      00034600
      00034700
      00034800
      00034900
      00035000
      00035100
      00035200
      00035300
      00035400
      00035500
      00035600
      00035700
      00035800
      00035900
      00036000
      00036100
      00036200
      00036300
      00036400
      00036500
      00036600
      00036700
      00036800
      00036900
      00037000
      00037100
      00037200
      00037300
      00037400
      00037500
      00037600
      00037700
      00037800
      00037900
      00038000
      00038100
      00038200
      00038300
      00038400
      00038500
      00038600
      00038700
      00038800
      00038900
      00039000
      00039100
      00039200
      00039300
      00039400
      00039500
      00039600
      00039700
      00039800
      00039900
      00040000
      00040100
      00040200
      00040300
      00040400
      00040500
      00040600
      00040700
      00040800
      00040900
      00041000
      00041100
      00041200
      00041300
      00041400
      00041500
      00041600
      00041700
      00041800
      00041900
      00042000
      00042100
      00042200
      00042300
      00042400
      00042500
      00042600
      00042700
      00042800
      00042900
      00043000
      00043100
      00043200
      00043300
      00043400
      00043500
      00043600
      00043700
      00043800
      00043900
      00044000
      00044100
      00044200
      00044300
      00044400
      00044500
      00044600
      00044700
      00044800
      00044900
      00045000
      00045100
      00045200
      00045300
      00045400
      00045500
      00045600
      00045700
      00045800
      00045900
      00046000
      00046100
      00046200
      00046300
      00046400
      00046500
      00046600
      00046700
      00046800
      00046900
      00047000
      00047100
      00047200
      00047300
      00047400
      00047500
      00047600
      00047700
      00047800
      00047900
      00048000
      00048100
      00048200
      00048300
      00048400
      00048500
      00048600
      00048700
      00048800
      00048900
      00049000
      00049100
      00049200
      00049300
      00049400
      00049500
      00049600
      00049700
      00049800
      00049900
      00050000
      00050100
      00050200
      00050300
      00050400
      00050500
      00050600
      00050700
      00050800
      00050900
      00051000
      00051100
      00051200
      00051300
      00051400
      00051500
      00051600
      00051700
      00051800
      00051900
      00052000
      00052100
      00052200
      00052300
      00052400
      00052500
      00052600
      00052700
      00052800
      00052900
      00053000
      00053100
      00053200
      00053300
      00053400
      00053500
      00053600
      00053700
      00053800
      00053900
      00054000
      00054100
      00054200
      00054300
      00054400
      00054500
      00054600
      00054700
      00054800
      00054900
      00055000
      00055100
      00055200
      00055300
      00055400
      00055500
      00055600
      00055700
      00055800
      00055900
      00056000
      00056100
      00056200
      00056300
      00056400
      00056500
      00056600
      00056700
      00056800
      00056900
      00057000
      00057100
      00057200
      00057300
      00057400
      00057500
      00057600
      00057700
      00057800
      00057900
      00058000
      00058100
      00058200
      00058300
      00058400
      00058500
      00058600
      00058700
      00058800
      00058900
      00059000
      00059100
      00059200
      00059300
      00059400
      00059500
      00059600
      00059700
      00059800
      00059900
      00060000
      00060100
      00060200
      00060300
      00060400
      00060500
      00060600
      00060700
      00060800
      00060900
      00061000
      00061100
      00061200
      00061300
      00061400
      00061500
      00061600
      00061700
      00061800
      00061900
      00062000
      00062100
      00062200
      00062300
      00062400
      00062500
      00062600
      00062700
      00062800
      00062900
      00063000
      00063100
      00063200
      00063300
      00063400
      00063500
      00063600
      00063700
      00063800
      00063900
      00064000
      00064100
      00064200
      00064300
      00064400
      00064500
      00064600
      00064700
      00064800
      00064900
      00065000
      00065100
      00065200
      00065300
      00065400
      00065500
      00065600
      00065700
      00065800
      00065900
      00066000
      00066100
      00066200
      00066300
      00066400
      00066500
      00066600
      00066700
      00066800
      00066900
      00067000
      00067100
      00067200
      00067300
      00067400
      00067500
      00067600
      00067700
      00067800
      00067900
      00068000
      00068100
      00068200
      00068300
      00068400
      00068500
      00068600
      00068700
      00068800
      00068900
      00069000
      00069100
      00069200
      00069300
      00069400
      00069500
      00069600
      00069700
      00069800
      00069900
      00070000
      00070100
      00070200
      00070300
      00070400
      00070500
      00070600
      00070700
      00070800
      00070900
      00071000
      00071100
      00071200
      00071300
      00071400
      00071500
      00071600
      00071700
      00071800
      00071900
      00072000
      00072100
      00072200
      00072300
      00072400
      00072500
      00072600
      00072700
      00072800
      00072900
      00073000
      00073100
      00073200
      00073300
      00073400
      00073500
      00073600
      00073700
      00073800
      00073900
      00074000
      00074100
      00074200
      00074300
      00074400
      00074500
      00074600
      00074700
      00074800
      00074900
      00075000
      00075100
      00075200
      00075300
      00075400
      00075500
      00075600
      00075700
      00075800
      00075900
      00076000
      00076100
      00076200
      00076300
      00076400
      00076500
      00076600
      00076700
      00076800
      00076900
      00077000
      00077100
      00077200
      00077300
      00077400
      00077500
      00077600
      00077700
      00077800
      00077900
      00078000
      00078100
      00078200
      00078300
      00078400
      00078500
      00078600
      00078700
      00078800
      00078900
      00079000
      00079100
      00079200
      00079300
      00079400
      00079500
      00079600
      00079700
      00079800
      00079900
      00080000
      00080100
      00080200
      00080300
      00080400
      00080500
      00080600
      00080700
      00080800
      00080900
      00081000
      00081100
      00081200
      00081300
      00081400
      00081500
      00081600
      00081700
      00081800
      00081900
      00082000
      00082100
      00082200
      00082300
      00082400
      00082500
      00082600
      00082700
      00082800
      00082900
      00083000
      00083100
      00083200
      00083300
      00083400
      00083500
      00083600
      00083700
      00083800
      00083900
      00084000
      00084100
      00084200
      00084300
      00084400
      00084500
      00084600
      00084700
      00084800
      00084900
      00085000
      00085100
      00085200
      00085300
      00085400
      00085500
      00085600
      00085700
      00085800
      00085900
      00086000
      00086100
      00086200
      00086300
      00086400
      00086500
      00086600
      00086700
      00086800
      00086900
      00087000
      00087100
      00087200
      00087300
      00087400
      00087500
      00087600
      00087700
      00087800
      00087900
      00088000
      00088100
      00088200
      00088300
      00088400
      00088500
      00088600
      00088700
      00088800
      00088900
      00089000
      00089100
      00089200
      00089300
      00089400
      00089500
      00089600
      00089700
      00089800
      00089900
      00090000
      00090100
      00090200
      00090300
      00090400
      00090500
      00090600
      00090700
      00090800
      00090900
      00091000
      00091100
      00091200
      00091300
      00091400
      00091500
      00091600
      00091700
      00091800
      00091900
      00092000
      00092100
      00092200
      00092300
      00092400
      00092500
      00092600
      00092700
      00092800
      00092900
      00093000
      00093100
      00093200
      00093300
      00093400
      00093500
      00093600
      00093700
      00093800
      00093900
      00094000
      00094100
      00094200
      00094300
      00094400
      00094500
      00094600
      00094700
      00094800
      00094900
      00095000
      00095100
      00095200
      00095300
      00095400
      00095500
      00095600
      00095700
      00095800
      00095900
      00096000
      00096100
      00096200
      00096300
      00096400
      00096500
      00096600
      00096700
      00096800
      00096900
      00097000
      00097100
      00097200
      00097300
      00097400
      00097500
      00097600
      00097700
      00097800
      00097900
      00098000
      00098100
      00098200
      00098300
      00098400
      00098500
      00098600
      00098700
      00098800
      00098900
      00099000
      00099100
      00099200
      00099300
      00099400
      00099500
      00099600
      00099700
      00099800
      00099900
      00100000
      00100100
      00100200
      00100300
      00100400
      00100500
      00100600
      00100700
      00100800
      00100900
      00101000
      00101100
      00101200
      00101300
      00101400
      00101500
      00101600
      00101700
      00101800
      00101900
      00102000
      00102100
      00102200
      00102300
      00102400
      00102500
      00102600
      00102700
      00102800
      00102900
      00103000
      00103100
      00103200
      00103300
      00103400
      00103500
      00103600
      00103700
      00103800
      00103900
      00104000
      00104100
      00104200
      00104300
      00104400
      00104500
      00104600
      00104700

```

DATE 84/03/16 (FRIDAY) TIME 13:39:11 PAGE 000

```

XX=96.+FLOAT(JM-1)*40.          00017100
XX0=XX-11.                      00017200
XX1=XX-11.                      00017300
XX2=XX-13.                      00017400
JJJ=JJ+JM-K                     00017500
CALL GSCHAR(XX0,205.,3,ITCM(1,JM),211,12) 00017600
DO 21 I=1,3                      00017700
II=I+3                          00017800
21 READ(ITCM(II,JM),6) MD1(I)    00017900
CALL GSCHAR(XX0,201.,3,MD1(1),211,12) 00018000
CALL GSCHAR(XX,193.,3,NSDIRC(1,JJJ),211,4) 00018100
CALL GSCHAR(XX1,186.,3,'ATOM DENSITY',211,12) 00018200
CALL GSCHR1(XX2,182.,3,'(&N/BARN.CM#)',211,13,99) 00018300
19 CONTINUE                      00018400
C
C      SET ATOMIC NUMBER DENSITY
DO 22 JM=1,K                     00018500
XX=88.+FLOAT(JM-1)*40.          00018600
NTMAX=NMC(JM)                   00018700
DO 23 JN=1,NTMAX                00018800
YY=172.-FLOAT(JN-1)*7.          00018900
DO 24 JO=1,NTMAX                00019000
LLLL=MID(JO,JM)                 00019100
AAAAA=AD(JO,JM)                 00019200
IF(MPC(JN).NE.LLLL) GO TO 24   00019300
WRITE(6,606) JN,JO,JM,AAAAA     00019400
606 FORMAT(5X,'MPC('',I3,'') MID('',I3,'',',I3,'') AD=',1PE12.5) 00019500
CALL NUMBRP(XX,YY,3,AAAAA,5,ICONT2,211) 00019600
00019700
00019800
00019900
00020000
00020100
00020200
00020300
00020400
00020500
00020600
00020700
00020800
00020900
00021000
00021100
00021200
00021300
00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500

C
C      MAKING FRAME
CALL PLOT(40.,210.,3)           00020400
CALL PLOT(280.,210.,2)           00020500
CALL PLOT(280.,40.,2)            00020600
CALL PLOT(40.,40.,2)             00020700
CALL PLOT(40.,210.,2)            00020800
CALL PLOT(40.,200.,3)            00020900
CALL PLOT(280.,200.,2)           00021000
CALL PLOT(40.,190.,3)            00021100
CALL PLOT(280.,190.,2)           00021200
CALL PLOT(40.,180.,3)            00021300
CALL PLOT(280.,180.,2)           00021400
CALL PLOT(80.,40.,3)              00021500
CALL PLOT(80.,210.,2)             00021600
CALL PLOT(120.,40.,3)             00021700
CALL PLOT(120.,210.,2)            00021800
CALL PLOT(160.,40.,3)             00021900
CALL PLOT(160.,210.,2)            00022000
CALL PLOT(200.,40.,3)             00022100
CALL PLOT(200.,210.,2)            00022200
CALL PLOT(240.,40.,3)             00022300
CALL PLOT(240.,210.,2)            00022400
CALL PLOT(240.,210.,2)            00022500

```

DATE 84/03/16 (FRIDAY) TIME 13:39:11 PAGE 000

Fig. 4.1 (continued)

DATE 84/03/16 (FRIDAY) TIME 13:39:11 PAGE 000

```

C          00022600
C          00022700
C          00022800
C          00022900
C          00023000
C          00023100
C          00023200
C          00023300
C TABLE OF CONTENTS IS PRODUCED BELOW 00023400
6000 NMAX=IMAX/21+1 00023500
JJ=0 00023600
NDPT=3 00023700
IF(NODE(1).EQ.NULTX) NDPT=2 00023800
DO 6010 M=1,NMAX 00023900
DO 6020 K=1,21 00024000
IF(JJ.GE.IMAX) GO TO 6200 00024100
JJ=JJ+1 00024200
WRITE(NODE(NDPT),6) NSDIRC(1,JJ)
CALL PDGET(IN,NODE,NDPT,JMAX,MSDIRC)
MMID(K)=NODE(NDPT)
MD1(K)=JMAX
DO 6014 I=1,JMAX 00024400
6014 LID(I,K)=MSDIRC(1,I) 00024500
CALL PFIND(IN,NODE,NDPT,ND,LLL) 00024600
IF(LLL.EQ.0) GO TO 6013 00024700
WRITE(6,3) LLL,IN,NODE(NDPT) 00024800
GO TO 1000 00024900
6013 CONTINUE 00025000
CALL PREAD(IN,NAME1,NAME2,ICM,JL1,JL2,JL3,DD) 00025100
DO 6015 I=1,6 00025200
II=I+1 00025300
IF(NODE(2).EQ.NSELF) II=I 00025400
6015 READ(ICM(II),6) ITCM(I,K) 00025500
READ(ICM(2),8) NTEMPA 00025600
NTEMPB=' '
IF(NODE(2).NE.NSELF) WRITE(ITCM(1,K),9) NTEMPB,NTEMPA 00025700
00025800
6020 CONTINUE 00025900
K=22 00026000
6200 K=K-1 00026100
IF(K.LE.0) GO TO 6010 00026200
CALL GSCHAR(50.,212.,3,JCONTR(1),211,80) 00026300
CALL GSCHAR(47.,203.,3,' NODE NAME',211,10) 00026400
CALL GSCHAR(90.,203.,3,'NUCLIDE NAME',211,12) 00026500
IF(NODE(2).NE.NSGRX) CALL GSCHAR(166.,203.,3,'CONTENTS',211,8) 00026600
IF(NODE(2).EQ.NSGRX) CALL GSCHAR(153.,203.,3,'REACTION CHANNEL', 00026700
+ 211,16) 00026800
DO 6021 I=1,K 00026900
MTEMP=MD1(I)*4 00027000
YY=193.-FLOAT(I-1)*7. 00027100
CALL GSCHAR(57.,YY,3,MMID(I),211,4) 00027200
CALL GSCHAR(77.,YY,3,ITCM(1,I),211,24) 00027300
IF(NODE(2).EQ.NSELF) GO TO 6022 00027400
CALL GSCHAR(143.,YY,3,LID(1,I),211,MTEMP) 00027500
GO TO 6021 00027600
6022 CONTINUE 00027700
LLML=MD1(I) 00027800
00027900
00028000

```

DATE 84/03/16 (FRIDAY) TIME 13:39:11 PAGE 000'

Fig. 4.1 (continued)

DATE 84/03/16(FRIDAY) TIME 13:39:11 PAGE 001

```

DO 6023 LG=1,LLML          00028100
XX=143.+FLOAT(LG-1)*12.    00028200
CALL GSCHAR(XX,YY,3,LID(LG,I),211,4) 00028300
6023 CONTINUE               00028400
6021 CONTINUE               00028500
C   MAKING FRAME             00028600
XXXX=210.                   00028700
IF(NODE(2).EQ.NSELF) XXXX=265. 00028800
CALL PLOT(47.,210.,3)        00028900
CALL PLOT(XXXX,210.,2)       00029000
CALL PLOT(XXXX,45.,2)        00029100
CALL PLOT(47.,45.,2)         00029200
CALL PLOT(47.,210.,2)        00029300
CALL PLOT(47.,200.,3)        00029400
CALL PLOT(XXXX,200.,2)       00029500
CALL PLOT(75.,45.,3)         00029600
CALL PLOT(75.,210.,2)        00029700
CALL PLOT(140.,45.,3)        00029800
CALL PLOT(140.,210.,2)       00029900
CALL PLOT(0.,0.,444)         00030000
CALL PLOT(0.,0.,666)         00030100
6010 CONTINUE               00030200
GO TO 1                     00030300
6 FORMAT(A4)                00030400
8 FORMAT(1X,A3)              00030500
9 FORMAT(A1,A3)              00030600
1000 STOP                    00030700
E N D                      00030800

```

\*\*\*\*\*  
\*\* TABLEJCL \*\*  
\*\*\*\*\*

```

//JCLG JOB
// EXEC JCLG
//SYSIN DD DATA,DLM='++'
// JUSER XXXXJUSERXX.XXXXXXX,XXXX.XXX
T.1 C.3 W.1 I.3 P.0 OPN      GRP
OPTP PASSWORD=??,NOTIFY=JUSER
// EXEC FORT77,SO='J3679.TABLE',A='ELM(*),SOURCE'
// EXEC LKED77,GRLIB=PNL,PRVLIB='J3679.DPOOL2'
// EXEC GO
// EXPAND GRNLP,SYSPUT=H
/* DATA-POOL IS SPECIFIED BY THE NEXT CARD
//FT91F001 DD DSN=J3679.FNSPOOL.DATA,DISP=SHR
/* # THE FOLLOWING NODE NAMES ARE ALLOWABLE #
/* THE ULTRA-FINE GROUP CROSS SECTIONS : ULTX
/* THE SECONDARY GAMMA-RAY PRODUCTION : EGRPSGRX
/* THE INFINITE DILUTION CROSS SECTIONS : EGRPINFX
/* THE MACROSCOPIC GROUP CROSS SECTIONS : EGRPFX32
/* THE SELF-SHIELDING FACTORS        : EGRPSELF
//SYSIN DD *
 91EGRPFX16     1     1
++
//
```

Fig. 4.1 (continued)

## 5. TSS Management Utility of DATA-POOL

### 5.1 Outline of POOL

The DATA-POOL is a special direct-access data base, so that the management utility POOL is prepared to maintain the data set. POOL has 14 operations commands and can be executed by TSS terminals or batch jobs.

On the TSS terminal, the next operation is necessary to execute the utility POOL. The statement with underline shows the user's input on TSS terminal. [CR] means a carriage return.

READY

EX 'J1446.TSSMAC.CLIST (POOLX)' [CR]

The user who has a data set named TSSMAC.CLIST(POOLX) can execute POOL by a simple operation. The "TSSMAC.CLIST" is a default data-set name consisted of user's cataloged procedures in the JAERI Computing Center.

READY

POOLX [CR]

The cataloged procedure is excuted, the next display appears.

READY

POOLX

FILE B NOT FREED, IS NOT ALLOCATED  
FILE DCB NOT FREED, IS NOT ALLOCATED  
FILE FT01F001 NOT FREED, IS NOT ALLOCATED  
FILE FT02F001 NOT FREED, IS NOT ALLOCATED  
FILE FT91F001 NOT FREED, IS NOT ALLOCATED  
\*\*\*\*\* STARTS RADHEAT-V4 DATA POOL UTILITY \*\*\*\*\*

ENTER COMMAND NAME ==>

The 12 functions of POOL can be selected by the user to enter a command name. HELP command is prepared when the user forgets command names. END command is used for terminating the execution. HELP command is entered, the next display appears.

ENTER COMMAND NAME ==> HELP

COMMAND	CONTENTS
CATL	PRINT OF CONTROL AND DIRECTORY SECTION
CONDENSE	CONDENSE OF A DATA POOL
COPY	COPY OF A NODE DATA
DELETE	DELETE OF A NODE DATA
FLAG	CHANGE OF A WRITE FLAG
INIT	INITIALIZATION OF A DATA POOL
LIST	LISTING OF A NODE DATA (SUB-DIRECTORY AND FORM OF DATA ARRAYS)
MEND	MENDING OF A CONTROL, DIRECTORY AND DATA COMMENT
MTCOPY	LOAD OF A BACK-UP TAPE TO A DATA POOL
MTSAVE	MAKING OF A BACK-UP TAPE
RENAME	RENAME OF A NODE
TREE	PRINT OF ALL NODE NAMES IN A DATA POOL BY A TREE STRUCTURE

ENTER COMMAND NAME ==>

Some commands have the abbreviated forms as follows :

FLAG	→ F
DELETE	→ DEL
TREE	→ T
CATL	→ C
LIST	→ L
RENAME	→ RE
CONDENSE	→ COND

The functions and usages of these commands are described in the following Sections. The operating method for batch job is described in Section 5.14. The cataloged procedure is shown in Section 5.15. The program information of the POOL is noted in Section 5.16.

## 5.2 INIT Command

DATA-POOL is initialized by using the INIT command. The initialization is executed by the conversational data inputs with TSS terminal. The initialization for the existing DATA-POOL means erasing the all of data from the DATA-POOL. The sample procedure is as follows :

ENTER COMMAND NAME ==> INIT

ENTER DSN OF DATA POOL ==> J3679.DATAPPOOL.DATA  
 ALLOCATION OF DATA SET (NEW/OLD) ==> NEW  
 UNIT PARAMETER ======> TSSWK  
 SPACE PARAMETER (1-ST SPACE) ======> 50  
 SPACE PARAMETER (INCREMENT) ======> 10  
 SPACE PARAMETER (SPACE UNIT T/CY) ==> T  
 ENTER DIRECTORY SIZE  
 01100 ?

20

ENTER TITLE (64 CHARACTERS)  
 01500 ?

RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE

\*\*\* MESSAGE FROM PINIT \*\*\*

NO. OF INITIALIZED RECORD IS 1000

\*\*\*\*\* C O N T R O L S E C T I O N \*\*\*\*\*

COL.

1-18 TITLE :

RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE

21	ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE	:	2
22	HEAD ADDRESS FOR THE VACANT DIRECTORY AREA	:	3
23	HEAD ADDRESS FOR THE VACANT DATA AREA	:	22
24	WRITE FLAG	:	0
25	READ FLAG (NOT USED)	:	0
26	LENGTH OF THE ONE PHYSICAL RECORD	:	900
27	MAXIMUM NUMBER OF THE SAME LEVEL NODE	:	74
28	SIZE OF THE DIRECTORY SECTION	:	20
29	SIZE OF THE DATA SECTION	:	979
30	REAL NUMBER OF THE DIRECTORY RECORDS	:	1
31	REAL NUMBER OF THE DATA SET RECORDS	:	0

ERROR SUMMARY (FORTRAN77)

ERROR NUMBER ERROR COUNT

232 001

In the sample, the UNIT PARAMETER means the unit name defined in a DD statement such as D0950B, TDS, TSSWK, MSS. The SPACE UNIT of T(track=19k bytes) or CY(Cylinder=250k bytes) can be specified. After the specifications for the data-set, two input data of a directory size and a comment (64-characters) are required. When the initialization is terminated, the Control Section is displayed and the initialized records can be known. An error of the error number 232 is prearranged one, so that the user is not necessary to pay attention. The data-set name must be specified as a full name because the abbreviated form may cause an error. Note that the volume number and the group number are also required when the allocation for MSS is specified. The allocation space for MSS may be a multiple of cylinder as follows :

ENTER COMMAND NAME ==> INIT

```
ENTER DSN OF DATA POOL ==> J3679.MSSDPPOOL.DATA
ALLOCATION OF DATA SET (NEW/OLD) ==> NEW
UNIT PARAMETER      ======> MSS
SPACE PARAMETER (1-ST SPACE)  ======> 10
SPACE PARAMETER (INCREMENT)   ======> 2
SPACE PARAMETER (SPACE UNIT T/CY) ==> CY
MSS GROUP          ======> MSS04
MSS VOLUME NUMBER    ======> MA0072
```

### 5.3 FLAG Command

The control flag for the exclusive access of the DATA-POOL in the Control Section is set zero by using the FLAG command as follows :

ENTER COMMAND NAME ==> FLAG

```
ENTER DSN OF DATA POOL ==> J3679.DATAPPOOL.DATA
CURRENT STATUS OF WRITE FLAG = 0
NOW CHANGE WRITE FLAG TO 0
```

### 5.4 DELETE Command

The node name and data are deleted by using the DELETE command.

A Sample is as follows :

ENTER COMMAND NAME ==> DELETE

```
ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA
ENTER NODE NAME
00900 ?
HA92.SELF.FEE4
NORMAL RETURN *** NODE NAME = HA92.SELF.FEE4.
ENTER NODE NAME
00900 ?
BAD.NODE.NAME
ABNORMAL RETURN *** NODE NAME = BAD.NODE.NAME.
ENTER NODE NAME
00900 ?
/*
```

The node name to be delete must be specified by the form such as "NOD1.NOD2.NOD3.NOD4" from the first column. A period between two node names must be specified. All directories and data with the last node name, and the all lower levels of this last node name specified by the user are erased. When the process is successfully ended, "NORMAL RETURN" is displayed. However, the process is not ended completely, "ABNORMAL RETURN" is displayed.

To terminate the process, /\* **[CR]** or **[CR]** should be entered.

### 5.5 TREE Command

The TREE command displays tree structures of node names as follows :

ENTER COMMAND NAME ==> TREE

ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA

```

N O D E   T R E E
TITLE OF A DATA POOL      ***
                    RADHEAT-V4 DATA-POOL FOR SKYSHINE CALCULATION
LENGTH OF A RECORD      ***      900
MAXIMUM NUMBER OF THE SAME LEVEL NODE ***      74
SIZE OF THE DIRECTORY SECTION      ***      40 (USED RECORDS 10)
SIZE OF THE DATA SECTION      ***      928 (USED RECORDS 391)
REMAINS OF THE DIRECTORY SECTION      ***      29
REMAINS OF THE DATA SECTION      ***      534

LEVEL     1      2      3      4      5      6      7      8

NAME : ENERGY GROUP STRUCTURE
|
I-----INFX : INFINITE DILUTION CROSS SECTION LIBRARY
|
I-----1276 : 1276 0 FROM ENDF/B-IV (300K)
|
I----- SMT : SMOOTH CROSS SECTIONS
I----- FTB : F-TABLE LIBRARY
I----- ELA : ELASTIC SCATTERING MATRIX
I----- INS : INELASTIC SCATTERING MATRIX

G09 : ENERGY GROUP STRUCTURE
|
I-----TEST : TEST GAMMA-RAY SKYSHINE
|
I-----SFX2 : TEST GAMMA-RAY SKYSHINE
I-----AFX2 : TEST GAMMA-RAY SKYSHINE

HA92 : ENERGY GROUP STRUCTURE
|
I-----SELF : SELF-SHIELDING FACTOR
I-----FX16 : ANGULAR MESH
|
I-----FEE4 : IRON ENDF/B-IV MACRO X-SEC. 92G
|
I-----1010 : NO. 101 IRON (0.9MFP) SPHERE (30DEG)
|
I-----SFX0 : NO. 101 IRON (0.9MFP) SPHERE (30DEG)

```

## 5.6 CATL Command

The CATL command displays the information of the Directory Section. The sample shown below is corresponded to the node structures of the previous Section. In the sample, "INDEX" means the address of physical record. The node name shown as "////" means that the directory was erased. The other variables are referred to the Section 2.

ENTER COMMAND NAME ==> CATL

ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA

D I R E C T O R Y L I S T							
***** C O N T R O L S E C T I O N *****							
COL.							
1-18	TITLE :	RADHEAT-V4 DATA-POOL FOR SKYSHINE CALCULATION					
21	ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE	:	2				
22	HEAD ADDRESS FOR THE VACANT DIRECTORY AREA	:	13				
23	HEAD ADDRESS FOR THE VACANT DATA AREA	:	436				
24	WRITE FLAG	:	0				
25	READ FLAG (NOT USED)	:	0				
26	LENGTH OF THE ONE PHYSICAL RECORD	:	900				
27	MAXIMUM NUMBER OF THE SAME LEVEL NODE	:	74				
28	SIZE OF THE DIRECTORY SECTION	:	40				
29	SIZE OF THE DATA SECTION	:	928				
30	REAL NUMBER OF THE DIRECTORY RECORDS	:	10				
31	REAL NUMBER OF THE DATA SET RECORDS	:	391				
***** D I R E C T O R Y S E C T I O N *****							
*** INDEX = 2 ***							
NODE NAME =							
ADDRESS FOR THE UPPER NODE DIRECTORY = 0							
NUMBER OF THE LOWER NODE = 3							
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	NAGE	1	3	42	1	84-01-20	
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)	
		102	0	0	0	0	
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
2	G09	1	6	134	1	84-01-24	
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)	
		0	9	0	0	0	
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
3	HA92	1	8	291	1	84-01-26	
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)	
		92	0	0	0	0	
*** INDEX = 3 ***							
NODE NAME = NAGE							
ADDRESS FOR THE UPPER NODE DIRECTORY = 2							
NUMBER OF THE LOWER NODE = 1							
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	INFX	1	4	43	1	84-01-20	
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)	
		0	0	0	0	0	

\*\*\* INDEX = 4 \*\*\*

NODE NAME = INFX  
 ADDRESS FOR THE UPPER NODE DIRECTORY = 3  
 NUMBER OF THE LOWER NODE = 1

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	1276	1	5	44	1	84-01-20	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		1276	0	0	0	0	0

\*\*\* INDEX = 5 \*\*\*

NODE NAME = 1276  
 ADDRESS FOR THE UPPER NODE DIRECTORY = 4  
 NUMBER OF THE LOWER NODE = 4

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	SMT	3	0	45	2	84-01-20	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		0	0	0	0	0	0
2	FTB	4	0	48	2	84-01-20	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		0	0	0	0	0	0
3	ELA	57	0	52	12	84-01-20	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		18	0	0	0	0	0
4	INS	25	0	109	12	84-01-20	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		0	0	0	0	0	0

\*\*\* INDEX = 6 \*\*\*

NODE NAME = G09  
 ADDRESS FOR THE UPPER NODE DIRECTORY = 2  
 NUMBER OF THE LOWER NODE = 1

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	TEST	2	7	135	1	84-01-24	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		5	41	29	2	48	

\*\*\* INDEX = 7 \*\*\*

NODE NAME = TEST  
 ADDRESS FOR THE UPPER NODE DIRECTORY = 6  
 NUMBER OF THE LOWER NODE = 2

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE	
1	SFX2	18	0	137	9	84-01-24	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		0	9	0	0	0	0
2	AFX2	136	0	155	46	84-01-24	INFOM(5)
		INFOM(1)	INFOM(2)	INFOM(3)	INFOM(4)		
		0	9	0	48	48	5

\*\*\* INDEX = 8 \*\*\*

	NODE NAME = HA92					
	ADDRESS FOR THE UPPER NODE DIRECTORY = 2					
	NUMBER OF THE LOWER NODE = 3					
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
1	SELF	1	9	292	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		0	0	0	0	0
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
2	FX16	1	11	296	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		16	0	0	0	0
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
3	1010	1	12	434	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		3	9	1	2	16

\*\*\* INDEX = 9 \*\*\*

	NODE NAME = SELF					
	ADDRESS FOR THE UPPER NODE DIRECTORY = 8					
	NUMBER OF THE LOWER NODE = 0					

\*\*\* INDEX = 10 \*\*\*

	NODE NAME = ////					
	ADDRESS FOR THE UPPER NODE DIRECTORY = 9					
	NUMBER OF THE LOWER NODE = 2					
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
1	1192	1	0	294	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		FEE4	2	4	0	0
NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
2	1274	1	0	295	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		FEE4	2	4	0	0

\*\*\* INDEX = 11 \*\*\*

	NODE NAME = FX16					
	ADDRESS FOR THE UPPER NODE DIRECTORY = 8					
	NUMBER OF THE LOWER NODE = 1					

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
1	FEE4	137	0	297	93	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		FEE4	4	3	95	0

\*\*\* INDEX = 12 \*\*\*

	NODE NAME = 1010					
	ADDRESS FOR THE UPPER NODE DIRECTORY = 8					
	NUMBER OF THE LOWER NODE = 1					

NO.	NODE	NRECS	NADWN	NADAT	NDASET	DATE
1	SFX0	1	0	435	1	84-01-26
	INFOM(1)		INFOM(2)	INFOM(3)	INFOM(4)	INFOM(5)
		92	0	0	0	0

## 5.7 LIST Command

The LIST command displays the record information for the node name specified by the user. A sample is as follows :

ENTER COMMAND NAME ==> LIST

ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA

ENTER NODE NAME

00900 ?

NAGE.INFX.1276

RECORD INFORMATION FOR NODE NAME NAGE.INFX.1276.

ITEM	CONTENTS
------	----------

1	NODE NAME	=	1276
2	TOTAL LENG. OF DATA SET	=	1
3	ADDRESS OF A LOWER NODE	=	5
4	ADDRESS OF A DATA SET	=	44
5	NO. OF SUB-DATA SETS	=	1
6	DATE OF CREATION	=	84-01-20
8	DATA 1	=	1276
9	DATA 2	=	0
10	DATA 3	=	0
11	DATA 4	=	0
12	DATA 5	=	0
<b>** INFORMATION FOR SUB-DATA SET 1 **</b>			
1276 0 FROM ENDF/B-IV (300K)			
NO. OF DATA ARRAYS = 1			
LENGTH OF DATA 1= 10 LENGTH OF DATA			

## 5.8 RENAME Command

The node names can be renamed by using the RENAME command. The renaming is performed to the last node name specified by the user. To terminate the process, /\*CR or CR should be entered.

A sample is as follows :

ENTER COMMAND NAME ==> RENAME

ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA

ENTER A OLD NODE NAME

00900 ?

HA92.SELF

ENTER A NEW NODE NAME

00900 ?

HA92.COOP

RENAME IS FINISHED SUCCESSFULLY

ENTER A OLD NODE NAME

00900 ?

/\*

## 5.9 COPY Command

The copy from a DATA-POOL to the other DATA-POOL can be performed by using the COPY command. When "\*ALL" is entered as a node name, all of data contained in a DATA-POOL are copied. If the node names are entered, data with the last level and the lower levels of nodes are copied. When the node name is already existed in the second data-set, the copy will not execute. The second data set should be initialized when the data set is newly created. To terminate the process, /\* [CR] or [CR] should be entered.

A sample is as follows :

```

ENTER COMMAND NAME ===> COPY

ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA
ENTER DSN OF 2-ND DATA POOL ===> J3679.DATAPPOOL.DATA
ENTER NODE NAME. IF *ALL IS ENTERD, ALL DATA IS COPIED
00900 ?
NAGE.INFX.1276
*** INFORMATION OF DATA POOL USAGE ***
LOGICAL UNIT NO.      =    92
DATA SET NAME          = J3679.DATAPPOOL.DATA
NO. OF WRITTEN RECORDS =    92
REMAINS RECORDS        =   887
DATA COPY WAS FINISHED SUCCESSFULLY
ENTER NEXT NODE NAME
00900 ?
G09.TEST
*** INFORMATION OF DATA POOL USAGE ***
LOGICAL UNIT NO.      =    92
DATA SET NAME          = J3679.DATAPPOOL.DATA
NO. OF WRITTEN RECORDS =   157
REMAINS RECORDS        =   730
DATA COPY WAS FINISHED SUCCESSFULLY
ENTER NEXT NODE NAME
00900 ?
/*

```

## 5.10 CONDENSE Command

The area for directories and data erased by the DELETE command remains as the unusable area, so that the release of the area should be performed by using the CONDENSE command. The condense procedure with TSS terminal or batch job can be chosen. In the case of TSS procedure, a sample is shown as follows :

ENTER COMMAND NAME ==> CONDENSE

```
EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ==> TSS
ENTER DSN OF DATA POOL ==> J3679.TEST00.DATA
BACK-UP DATA SET J3679.CONDENSE WAS CREATED
*** INFORMATION OF DATA POOL USAGE ***
LOGICAL UNIT NO. = 91
DATA SET NAME = J3679.TEST00.DATA
NO. OF WRITTEN RECORDS = 390
REMAINS RECORDS = 538
ENTRY (A) J3679.CONDENSE DELETED
```

In the case of batch job, the next Job Control Language is displayed.  
 A TSS terminal with FSO (Full Screen Option) may be needed to use the  
 option for batch job. The sample JCL is shown as follows :

ENTER COMMAND NAME ==> COND

EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ==> BATCH

```
EDIT-FSO (V01/L06) --- J3679.0POOLJCL.CNTL
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40 NONULLS 50
-----1----*---2----*---3----*---4----*---5----*---6----*---7
0010 //JCLG JOB
0020 //*****
0030 /* JOB CONTROL LANGUAGE FOR CONDENSE COMMAND *
0040 /*
0050 /* PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND *
0060 /* BACK-UP FILE NAME *
0061 /* AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND *
0070 /*
0080 //*****
0090 // EXEC JCLG
0100 //SYSIN DD DATA, DLM='++'
0110 // JUSER ???????, XX.XXXXXX, YYYY.ZZZ
0120 T.4 C.1 W.0 I.5 P.0 OPN
0130 OPTP PASSWORD=??
0140 // EXEC LMGO, LM='J3679.POOLX', PNM=COND
0150 /* DATA POOL
0160 /* CHANGE DSN:DATA SET NAME
0170 // EXPAND DISKTO, DDN=FT91F001, DSN='JXXXX.???????' , MODE=OUT
0180 /* BACK-UP FILE
0190 /* CHANGE DSN:DATA SET NAME
0220 // EXPAND DISKTN, DDN=FT01F001, DSN='JXXXX.00BACKUP' , UNIT=TSSWK,
0230 // SPC='500,300'
0240 // EXPAND DISK, DDN=FT02F001
0250 ++
0260 //
*** END OF DATA SET ***
```

The condensation starts after the backup of the data set is created, so  
 that the recovery of the data-set can be performed by using a MTCOPY command  
 when the condensation is abnormally terminated and the data-set is destroyed.  
 The description of the MTCOPY command is shown in Section 5.13.

## 5.11 MEND Command

The MEND command is prepared for recovery of the DATA-POOL when the data-set is destroyed. The user can change data in the Control, the Directory and the comment of the Data Sections, so that the revision of tree structures, information in the sub-directory and control variables are possible. The user should search the structure of data-linkage by using the CATL command before the MEND command is executed.

At the first stage of MEND execution, 3 options can be selected by the user. The option names are CONT, DIREC and COM. This command must be entered from the first column. The CONT option is used for the revision for the Control Section of the DATA-POOL. A sample is shown as follows :

```

ENTER COMMAND NAME ==> MEND

ENTER DSN OF DATA POOL ==> J3679.DATAPool.DATA
++ DATA POOL INFORMATION ++
TITLE :
      RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
1ST. RECORD NO. OF DIRECTORY :   2
LAST RECORD NO. OF DIRECTORY :   7
REAL RECORD NO. OF DIRECTORY :   6
WRITE PERMIT OF THE DATA POOL :   0
ENTER OPTION NAME CONT/DIREC/COM/END
00320 ?
CONT
***** C O N T R O L S E C T I O N *****

ITEM
1 TITLE :
      RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
21 ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE :   2
22 HEAD ADDRESS FOR THE VACANT DIRECTORY AREA :   8
23 HEAD ADDRESS FOR THE VACANT DATA AREA : 271
24 WRITE FLAG :   1
25 READ FLAG (NOT USED) :   0
26 LENGTH OF THE ONE PHYSICAL RECORD : 900
27 MAXIMUM NUMBER OF THE SAME LEVEL NODE :   74
28 SIZE OF THE DIRECTORY SECTION :   20
29 SIZE OF THE DATA SECTION : 979
30 REAL NUMBER OF THE DIRECTORY RECORDS :   6
31 REAL NUMBER OF THE DATA SET RECORDS : 249
ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS
00550 ?
1
ENTER NEW TITLE
00630 ?
DATA-POOL COPIED FROM J3679.TEST00.DATA
ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS
00550 ?
0

```

END OF MENDING A CONTROL SECTION SUCCESSFULLY  
\*\*\*\*\* C O N T R O L S E C T I O N \*\*\*\*\*

## ITEM

1	TITLE :	
DATA-POOL COPIED FROM J3679.TEST00.DATA		
21	ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE	: 2
22	HEAD ADDRESS FOR THE VACANT DIRECTORY AREA	: 8
23	HEAD ADDRESS FOR THE VACANT DATA AREA	: 271
24	WRITE FLAG	: 1
25	READ FLAG (NOT USED)	: 0
26	LENGTH OF THE ONE PHYSICAL RECORD	: 900
27	MAXIMUM NUMBER OF THE SAME LEVEL NODE	: 74
28	SIZE OF THE DIRECTORY SECTION	: 20
29	SIZE OF THE DATA SECTION	: 979
30	REAL NUMBER OF THE DIRECTORY RECORDS	: 6
31	REAL NUMBER OF THE DATA SET RECORDS	: 249

The user selects the item No. (1 ~ 31) to be change. The revised value or title should be entered next. These entries are given by a free format. The sequence is repeated until a [CR] or a 0 [CR] entry. The revised Control Section is displayed at the end of processing, and a next option is required. To terminate the MEND command, an END command may be entered.

The DIREC option is used to change the Directory Section. The user can change the sub-directory (SUB) and the head information defined in the directory (HEAD). A sample is shown below for the case of revision of the sub-directory.

```

ENTER OPTION NAME CONT/DIREC/COM/END
00320 ?
DIREC
ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS
00900 ?
NAGE, INFEX. 1276. SMT
ENTER OPTION NAME SUB/LOW
01040 ?
SUB
    ITEM      CONTENTS
    1  NODE NAME      =      SMT
    2  TOTAL LENG. OF DATA SET =      3
    3  ADDRESS OF A LOWER NODE =      0
    4  ADDRESS OF A DATA SET =      25
    5  NO. OF SUB-DATA SETS =      2
    6  DATE OF CREATION      =84-01-20
    8  DATA 1 =          0
    9  DATA 2 =          0
   10  DATA 3 =          0
   11  DATA 4 =          0
   12  DATA 5 =          0
ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTORY
IF ENTER 0, END TO MEND THE SUB-DIRECTORY
01210 ?

```

ENTER NEW VALUE  
01430 ?  
777  
INPUT VALUE WAS INTEGER TYPE 777  
ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTORY  
IF ENTER 0, END TO MEND THE SUB-DIRECTORY  
01210 ?  
0  
END OF MENDING A SUB-DIRECTORY SECTION SUCCESSFULLY  

ITEM	CONTENTS
1 NODE NAME	= SMT
2 TOTAL LENG. OF DATA SET	= 3
3 ADDRESS OF A LOWER NODE	= 0
4 ADDRESS OF A DATA SET	= 25
5 NO. OF SUB-DATA SETS	= 2
6 DATE OF CREATION	= 84-01-20
8 DATA 1	= 0
9 DATA 2	= 0
10 DATA 3	= 0
11 DATA 4	= 777
12 DATA 5	= 0

In the option, the node names to be revise must be entered such as "NOD1.NOD2.NOD3.NOD4" from the first column. The user selects the item No. (1~12) to be change or a DEL command from the first column to erase the sub-directory. When the item No. is entered, the new value must be entered next. The format is free but a real quantity should be less than 13 digits contained a decimal point. The process is repeated until [CR] or 0 [CR] is entered.

A sample is shown below when the option of "HEAD" is entered.

ENTER OPTION NAME CONT/DIREC/COM/END  
00320 ?  
DIREC  
ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS  
00900 ?  
NAGE. INFX. 1276  
ENTER OPTION NAME SUB/HEAD  
01040 ?  
HEAD  
DIRECTORY HEAD  

ITEM	CONTENTS
1 NODE NAME 1	1276
2 NODE NAME 2	
3 UPPER DIRECTORY ADDRESS	4
4 NO. OF SUB-DIRECTORY	4

NODE NAMES FOR EACH SUB-DIRECTORY  
SMT FTB ELA INS  
ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS DIRECTORY HEAD  
IF ENTER 0, END TO MEND THE DIRECTORY  
01890 ?  
DEL  
DIRECTORY HEAD1276 WAS DELETED  
ENTER OPTION NAME CONT/DIREC/COM/END  
00320 ?  
END  
END OF MEND COMMAND

The directory has the node names and the linkage information consisted of 4 variables at the first part of the each record. The user can change the tree structures of node names by using the "HEAD" option and setting 3 or 4 to the item No. If the user enters "DEL" from the first column, the directory and the linkage of the lower sub-directories is erased.

When the option of "COM" is selected, the comments of the sub-data sets are displayed. The user selects the sub-data set No. and enters a new comment (80 characters). The sequence is repeated until 0 [CR] or [CR] is entered. A sample is as follows :

```

ENTER OPTION NAME  CONT/DIREC/COM/END
00320 ?
COM
ENTER NODE NAME. IF ENTER NOTHING. END TO PROCESS
00900 ?
NAGE. INFEX
NO. OF SUB-DATA-SET IS  1
DAT NO.      COMMENT
    1  INFINITE DILUTION CROSS SECTION LIBRARY
ENTER DAT NO. TO MEND. IF ENTER 0. END TO PROCESS
02470 ?
1
ENTER NEW COMMENT
02540 ?
INFINITE DILUTION CROSS SECTIONS
ENTER DAT NO. TO MEND. IF ENTER 0, END TO PROCESS
02470 ?
0
END OF MENDING DATA COMMENTS SUCCESSFULLY
DAT NO.      COMMENT
    1  INFINITE DILUTION CROSS SECTIONS

```

The MEND command contains the complicated data entries, so that the user should take care to enter exact values. The operations flow of the MEND command is shown in Fig. 5.1.

### 5.12 MTSAVE Command

The MTSAVE command generates a Job Control Language to store data from the DATA-POOL to a backup tape or a sequential data-set. The user specifies the data-set names of the DATA-POOL and of MT, the volume serial number and the position, then enters an EDIT-mode command of "SUBMIT". A TSS terminal with FSO (Full Screen Option) may be needed to use the command. A sample JCL is shown as follows :

ENTER COMMAND NAME ==> MTSAVE

```

EDIT-FSO (V01/L06) --- J3679.0POOLJCL.CNTL
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40 NONULLS 50
-----1----2----3----4----5----6----7
0010 //JCLG JOB
0020 //***** JOB CONTROL LANGUAGE FOR MTSAVE COMMAND *
0030 //*
0040 //*
0050 //* PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND *
0060 //* BACK-UP TAPE NAME *
0061 //* AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND *
0070 //*
0080 //***** *
0090 // EXEC JCLG
0100 //SYSIN DD DATA, DLM='++'
0110 // JUSER ???????, XX. XXXXXX, YYYY. ZZZ
0120 T.4 C.1 W.0 I.5 P.0 OPN MTU
0130 OOPTP PASSWORD=??
0140 // EXEC LMGO, LM='J3679.POOLX', PNM=MTSAVE
0150 //* DATA POOL
0160 //* CHANGE DSN:DATA SET NAME
0170 // EXPAND DISKTO, DDN=FT91F001, DSN='JXXXX. ???????'
0180 //* BACK-UP TAPE
0190 //* CHANGE DSN:DATA SET NAME
0200 //* MTV:VOLUME NUMBER OF A TAPE
0210 //* POS:DATA SET POSITION ON A TAPE
0220 // EXPAND TAPE, DDN=FT01F001, DSN='JXXXX. ???????', MTV=??????, MTU=TAPE,
0230 // POS=?, DISP='NEW, PASS'
0240 // EXPAND DISK, DDN=FT02F001
0250 ++
0260 //
*** END OF DATA SET ***

```

### 5.13 MTCOPY Command

The MTCOPY command is prepared to recover data from a MT or a sequential data-set saved by using the MTSAVE command. The user specifies the data-set names of the DATA-POOL and of MT, the volume serial number and the position, then enters an EDIT-mode command of "SUBMIT". A sample JCL is shown as follows :

ENTER COMMAND NAME ==> MTCOPY

```

EDIT-FSO (V01/L06) --- J3679.0POOLJCL.CNTL
==>
ROW SCROLL ==> PAGE COLUMN SCROLL ==> 40 NONULLS 50
-----1----*---2---*---3---*---4---*---5---*---6---*---7
0010 //JCLG JOB
0020 //*****JOB CONTROL LANGUAGE FOR MTCOPY COMMAND *****
0030 //*      JOB CONTROL LANGUAGE FOR MTCOPY COMMAND *
0040 //*
0050 //*      PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND *
0060 //*          BACK-UP TAPE NAME *
0061 //*      AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND *
0070 //*
0080 //*****DATA SET NAME*****
0090 // EXEC JCLG
0100 //SYSIN DD DATA, DLM='++'
0110 // JUSER ???????, XX.XXXXXX, YYYY.ZZZ
0120 T.4 C.1 W.0 I.5 P.0 OPN MTU
0130 OPTP PASSWORD=?
0140 // EXEC LMGO, LM='J3679.POOLX', PNM=MTCOPY
0150 //* DATA POOL
0160 //* CHANGE DSN:DATA SET NAME
0170 // EXPAND DISKTO, DDN=FT91F001, DSN='JXXXX.????????', MODE=OUT
0180 //* BACK-UP TAPE
0190 //* CHANGE DSN:DATA SET NAME
0200 //* MTV:VOLUME NUMBER OF A TAPE
0210 //* POS:DATA SET POSITION ON A TAPE
0220 // EXPAND TAPE, DDN=FT01F001, DSN='JXXXX.????????', MTV=??????, MTU=TAPE,
0230 //    POS=?
0240 ++
0250 //
*** END OF DATA SET ***

```

Note that excess data more than 50,000 records can not be treated by using the MTCOPY and the MTSAVE commands. The DATA-POOL using in the MTCOPY command should be initialized before the batch job is submitted. The title in the Control Section will be replaced to the title in the backup tape.

#### 5.14 Operating Method of POOL for Batch Job

The load module of POOL is generated by each command name, so that the executions are performed by specifying the program name (PNM name).

// EXEC LMGO, LM='J3679.POOLX', PNM=program name  
Table 5.1 shows the program names and the input/output data-sets corresponded to the command names. The input data for each program are the same as the data entries described in previous Sections. The entry of [CR] can be

replaced by a blank card. The following cataloged procedure is useful to generate the load module of POOL.

### 5.15 TSS Cataloged Procedure

The cataloged procedure of POOL is shown as follows :

```
PROC 0
CONTROL PROMPT
FREE ATTRLIST(B)
FREE ATTRLIST(DCB)
ATTR B LRECL(133) RECFM(U A)
ATTR DCB LR(3600) BL(3600) REC(F)
FREE F(FT01F001)
FREE F(FT02F001)
FREE F(FT06F001)
FREE F(FT91F001)
ALLOC DA(*) F(FT06F001) USING(B)
WRITE ***** STARTS RADHEAT-V4 DATA POOL UTILITY *****
STAT:WRITENR ENTER COMMAND NAME ===>
READ &ELM
IF &ELM=INIT THEN GOTO JUMP1
IF &ELM=F      THEN SET &ELM=FLAG
IF &ELM=FLAG THEN GOTO JUMP2
IF &ELM=DEL   THEN SET &ELM=DELETE
IF &ELM=DELETE THEN GOTO JUMP2
IF &ELM=T      THEN SET &ELM=TREE
IF &ELM=TREE  THEN GOTO JUMP2
IF &ELM=C      THEN SET &ELM=CATL
IF &ELM=CATL  THEN GOTO JUMP2
IF &ELM=MEND  THEN GOTO JUMP2
IF &ELM=L      THEN SET &ELM=LIST
IF &ELM=LIST  THEN GOTO JUMP2
IF &ELM=RE     THEN SET &ELM=RENAME
IF &ELM=RENAME THEN GOTO JUMP2
IF &ELM=COPY  THEN GOTO JUMP4
IF &ELM=CONDENSE THEN SET &ELM=COND
IF &ELM=COND  THEN GOTO JUMP3
IF &ELM=MTSAVE THEN GOTO JUMP5
IF &ELM=MTCOPY THEN GOTO JUMP5
```

```
IF &ELM=HELP THEN GOTO RUN
IF &ELM=END THEN GOTO FINIS
WRITE ERROR COMMAND NAME. PLEASE RECONFIRM BY HELP COMMAND
GOTO STAT
JUMP1:WRITENR ENTER DSN OF DATA POOL ===>
READ &DSN
WRITENR ALLOCATION OF DATA SET (NEW/OLD) =====>
READ &ANS
IF &ANS=OLD THEN +
DO
  ALLOC F(FT91F001) DSN('&DSN') SHR
  GOTO RUN
END
WRITENR UNIT PARAMETER ======>
READ &UNIT
WRITENR SPACE PARAMETER (1-ST SPACE) =====>
READ &SPC
WRITENR SPACE PARAMETER (INCREMENT) =====>
READ &INC
WRITENR SPACE PARAMETER (SPACE UNIT T/CY) ===>
READ &T
IF &UNIT=MSS THEN GOTO JUMP11
ALLOC DA('&DSN') F(FT91F001) UNIT(&UNIT) SP(&SPC &INC) &T US(DCB) +
  NEW CAT
GOTO RUN
JUMP11:WRITENR MSS GROUP =====>
READ &MSVGP
WRITENR MSS VOLUME NUMBER =====>
READ &VOL
ALLOC DA('&DSN') F(FT91F001) UNIT(&UNIT) MSVGP(&MSVGP) VO(&VOL) +
  SP(&SPC &INC) &T US(DCB) NEW CAT
GOTO RUN
JUMP2:WRITENR ENTER DSN OF DATA POOL ===>
READ &DSN
ALLOC F(FT91F001) DSN('&DSN') SHR
GOTO RUN
JUMP3:WRITENR EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ===>
READ &TSS
```

```
IF &TSS=TSS THEN GOTO JUMP4
GOTO JUMP5
JUMP4:WRITENR ENTER DSN OF DATA POOL ===>
READ &DSN
ALLOC F(FT91F001) DSN('&DSN') SHR
IF &ELM=COPY THEN +
DO
  WRITENR ENTER DSN OF 2-ND DATA POOL ===>
  READ &DSN2
  ALLOC DA('&DSN2') F(FT92F001) SHR
  ALLOC DSN(CONDENSE) F(FT01F001) NEW SP(500 200) T UNIT(TSSWK) CAT
  ALLOC F(FT02F001) NEW SP(10 10) T UNIT(TSSWK)
GOTO RUN
JUMP5:COPY 'J3679.TSSMAC.CNTL(&ELM)' @POOLJCL.CNTL
E @POOLJCL.CNTL CN;FS
DEL @POOLJCL.CNTL
GOTO STAT
RUN:CALL 'J1446.POOLX.LOAD(&ELM)'
IF &ELM NE HELP THEN FREE F(FT91F001)
IF &ELM=COPY OR &ELM=COND THEN +
DO
  DEL CONDENSE
  FREE F(FT02F001)
END
TOTO STAT
FINIS:FREE ATTRLIST(B)
FREE ATTRLIST(DCB)
FREE F(FT06F001)
EXIT
```

Table 5.1 Program names to execute the commands of POOL

Command Name	PNM Name	I/O Files	Comment
INIT	INIT	FT91F001	DATA-POOL
FLAG	FLAG	FT91F001	DATA-POOL
DELETE	DELETE	FT91F001	DATA-POOL
TREE	TREE	FT91F001	DATA-POOL
CATL	CATL	FT91F001	DATA-POOL
MEND	MEND	FT91F001	DATA-POOL
LIST	LIST	FT91F001	DATA-POOL
RENAME	RENAME	FT91F001	DATA-POOL
COPY	COPY	FT91F001 FT92F001 FT01F001 <sup>+1</sup> FT02F001 <sup>+2</sup>	DATA-POOL to be read DATA-POOL to be written working file working file
CONDENSE	COND	FT91F001 FT01F001 <sup>+1</sup> FT02F001 <sup>+2</sup>	DATA-POOL working file working file
MTSAVE	MTSAVE	FT91F001 FT01F001 FT02F001 <sup>+2</sup>	DATA-POOL Magnetic Tape working file
MTCOPY	MTCOPY	FT91F001 FT01F001	DATA-POOL Magnetic Tape or Back-up data-set

+1 This working file is used to store data in DATA-POOL, so that the sufficient space may be required, and it is desirable to allocate the working file as a cataloged data-set on TSSWK unit. When the execution of COND or COPY procedure is terminated abnormally, the data are recoverable by using the data set with MTCOPY command.

- +2 This working file is used to store node names. The large space is not required.

These working files are ordinary sequential data-sets. The DCB information may be specified as LRECL=6208, BLKSIZE=6212, RECFM=VBS.

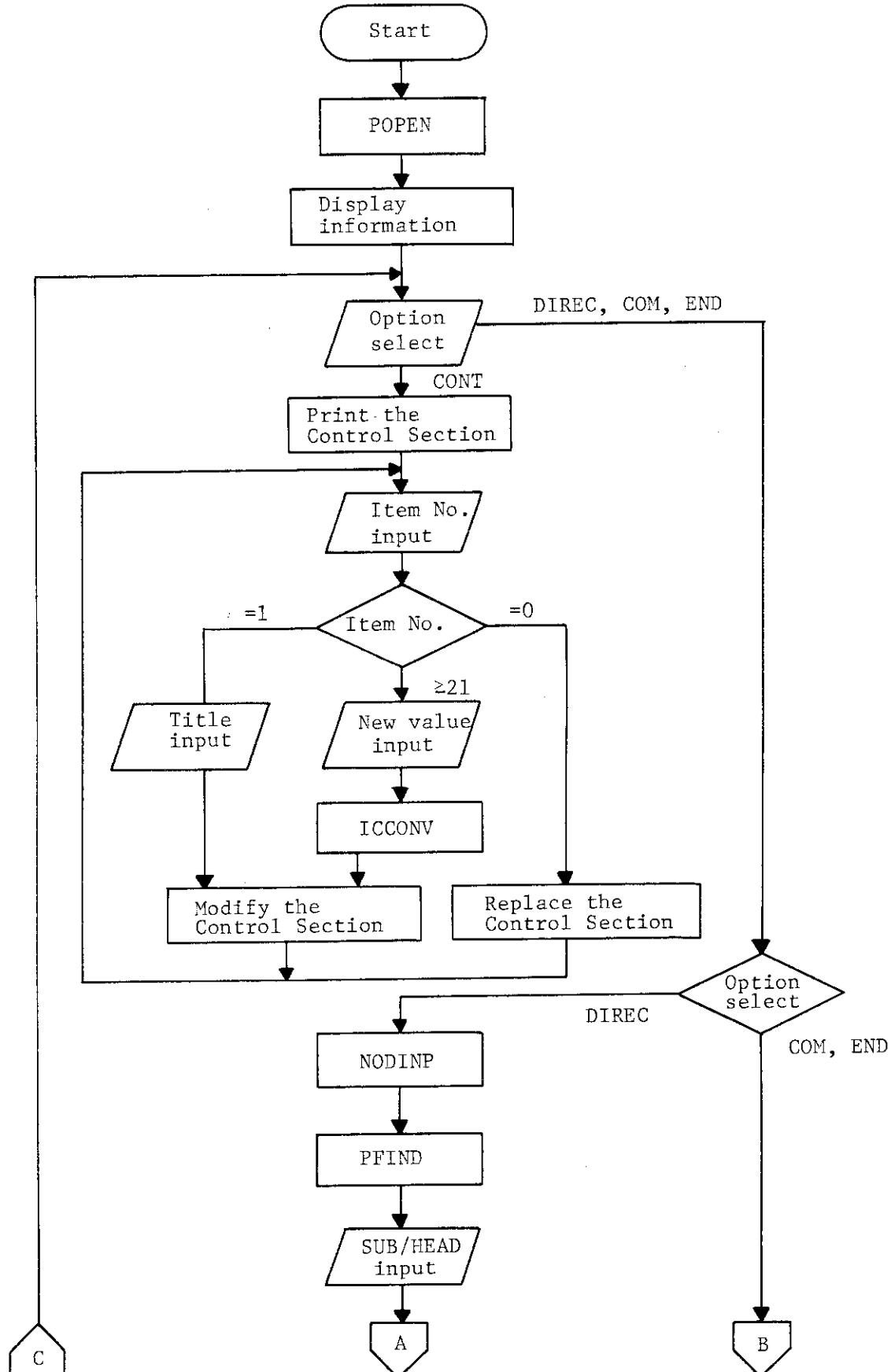


Fig. 5.1 Flow chart of MEND procedure

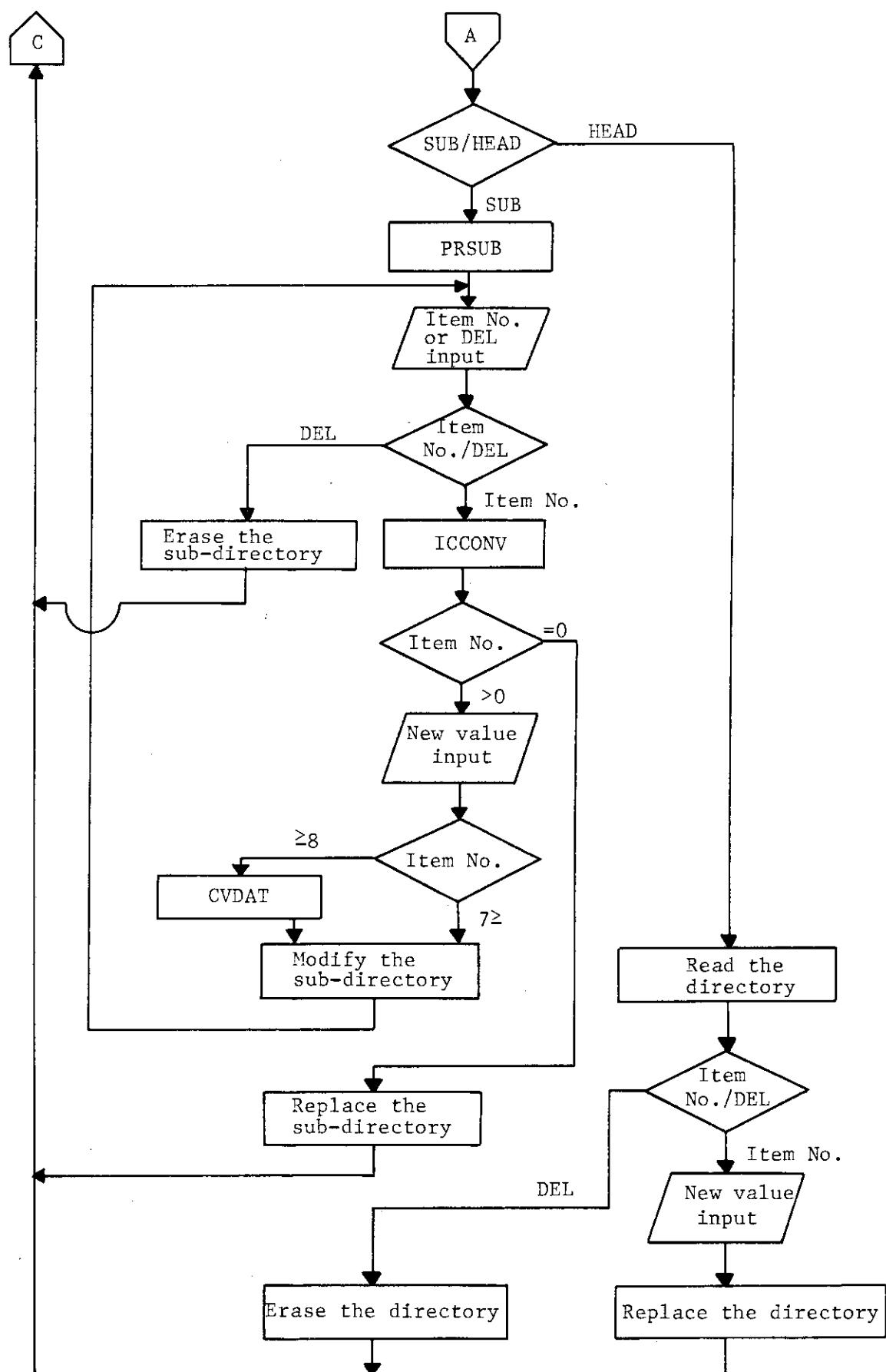


Fig. 5.1 (continued)

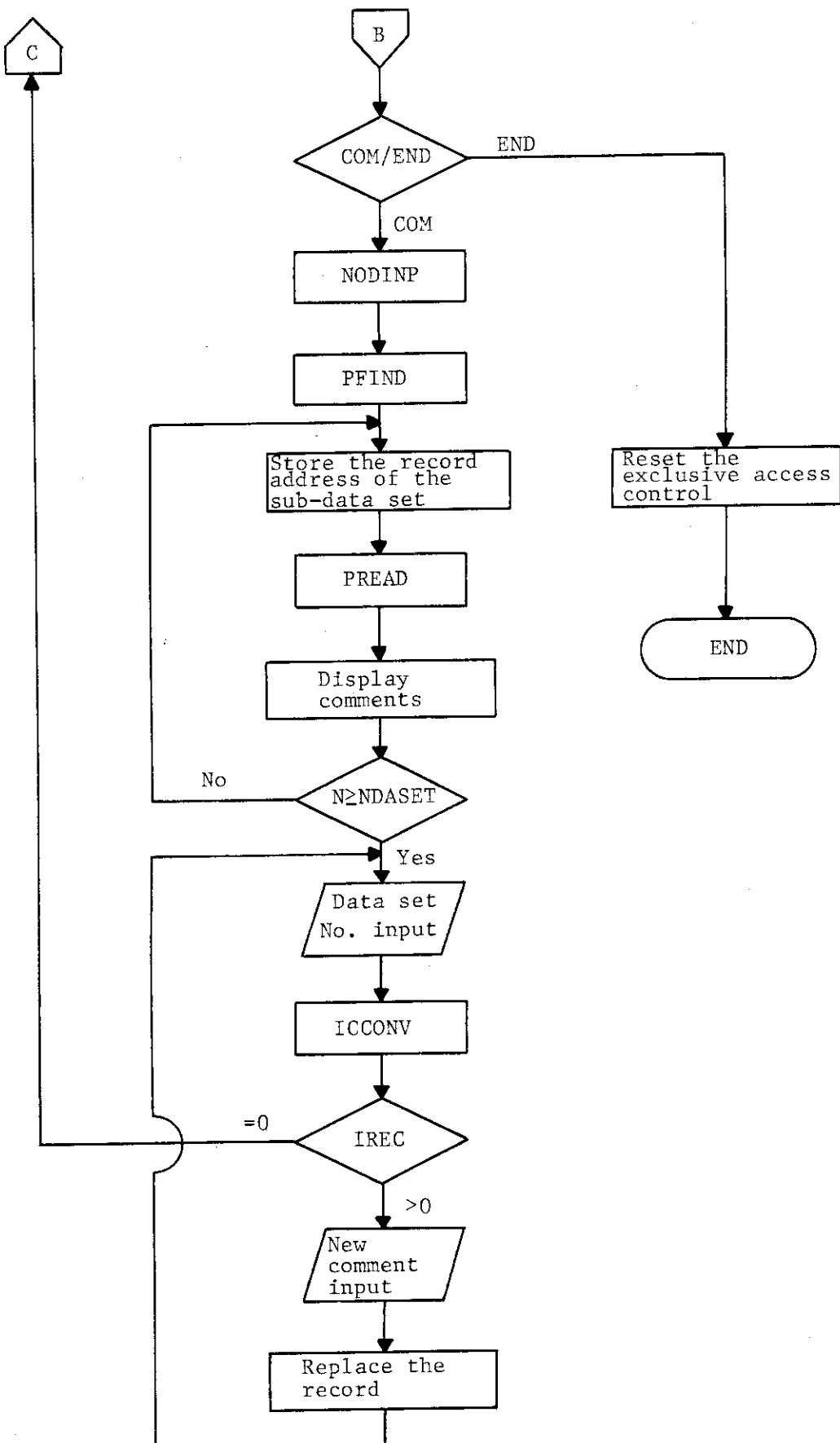


Fig. 5.1 (continued)

## 6. Record Format of DATA-POOL for RADHEAT-V4

The data processed in the RADHEAT-V4 code system are classified according to 8 subjects as shown in Fig. 6.1. The data in a subject relate to each other with the node tree structure. The node names and the node structure are described in this Section to display an application of the DATA-POOL for large-scale scientific computer code systems.

The data are mainly associated with the node of the last level, so that the data are classified and stored in the 17 forms described below. In the following description, the node name with a capital letter in the classification are used as the fixed name, and a small letter means that the name may change for each of data.

### a) ULTX Data Form

ULTX - matno - TMPi - SIGj

This form contains the ultra-fine group cross section. The identifications for the node names are as follows :

level 1 : ULTX shows the ultra-fine group data. The energy group structure is stored in the node.

level 2 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 3 : TMPi shows the temperature. The index i indicates the temperature of i. In RADHEAT-V4, i = 1, 2, 3, 4, 5 means 300, 560, 900, 1200 and 2100 K, respectively.

level 4 : SIGj shows the background cross section. The index j indicates the  $\sigma_0$  value. j = 1 means  $10^8$  in RADHEAT-V4.

### b) SMT Data Form

EGRP -INFX - matno - SMT

This form contains the smooth cross sections with the fine-group structure. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data. The energy group structure is stored in the node.

level 2 : INFX shows the infinitely diluted cross section.

level 3 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 4 : SMT shows the smooth cross section.

c) FTB Data Form

EGRP - INFX - matno - FTB

This form contains the self-shielding factors of each  $\sigma_0$  value. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name FTB indicates that the self-shielding factors are stored in the node.

d) ELA Data Form

EGRP - INFX - matno - ELA

This form contains the scattering matrix of elastic reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name ELA indicates that the scattering matrix of elastic reaction is stored in the node.

e) INS Data Form

EGRP - INFX - matno - INS

This form contains the scattering matrix of inelastic reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name INS indicates that the scattering matrix of inelastic reaction is stored in the node.

f) N2N Data Form

EGRP - INFX - matno - N2N

This form contains the scattering matrix of  $(n,2n)$  reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name N2N indicates that the scattering matrix of  $(n,2n)$  reaction is stored in the node.

## g) H+D Data Form

EGRP - INFX - matno - H+D

This form contains the energy deposition factor and atomic displacement cross section. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name H+D indicates that the energy deposition factor and the atomic displacement cross section are stored in the node.

## h) SGRX Data Form

EGRP - SGRX - matno - ncode

This form contains the secondary gamma-ray production cross sections of each reaction. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : SGRX shows the secondary gamma-ray production cross section.

level 3 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 4 : ncode shows the reaction channel.

## i) FXsn Data Form

EGRP - FXsn - matid

This form contains the effective macroscopic cross section. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : FXsn shows the effective macroscopic cross section and the number of angular points.

level 3 : matid shows the material name.

## j) SELF Data Form

EGRP - SELF - matid - matno

This form contains the self-shielding factors of each nuclide in the material defined by the matid in the FXsn data form. The data are utilized for generating the effective macroscopic cross section. The identifications for the node names are as follows :

- level 1 : EGRP shows the fine-group data same as the SMT data form.
- level 2 : SELF shows that the self-shielding factors of each nuclide are defined in the FXsn data form.
- level 3 : matid shows the material name defined in the FXsn data form.
- level 4 : matno shows the nuclide number contained in the material.

k) SFX0/SFX1 Data Form

EGRP - problem no.  SFX0  
SFX1

This form contains forward scalar fluxes (SFX0) and adjoint scalar fluxes (SFX1) generated by an one-dimensional SN-transport code DIAC. The data are used for generating few-group cross sections and for calculations of reaction rates. The identifications for the node names are as follows :

- level 1 : EGRP shows the fine-group data same as the SMT data form.
- level 2 : problem no. indicates the problem identification number specified by the first value of 15 \$ array in the input data of DIAC.
- level 3 : SFX0 means forward scalar fluxes of DIAC.  
SFX1 shows adjoint scalar fluxes of DIAC.

l) SFX2/SFX3 Data Form

EGFG - problem no.  SFX2  
SFX3

This form contains forward scalar fluxes (SFX2) and adjoint scalar fluxes (SFX3) generated by a two-dimensional SN-transport code ESPRIT. The data are used for calculations of reaction rates. The identifications for the node names are as follows :

- level 1 : EGFG shows the energy group structure used in the ESPRIT calculations.
- level 2 : problem no. indicates the problem identification number or name (4-characters) specified by the first data in the title card of ESPRIT.
- level 3 : SFX2 shows forward scalar fluxes of ESPRIT.  
SFX3 means adjoint scalar fluxes of ESPRIT.

## m) AFX0/AFX1 Data Form

EGRP - problem no. └── AFX0  
                          └── AFX1

This form contains forward angular fluxes (AFX0) and adjoint angular fluxes (AFX1) generated by the one-dimensional  $S_N$ -transport code DIAC.  
 The identifications for the node names are as follows :

- level 1 : EGRP means the energy group structure same as the SMT data form.
- level 2 : problem no. is the same as the SFX0/SFX1 data form.
- level 3 : AFX0 shows forward angular fluxes of DIAC.  
 AFX1 means adjoint angular fluxes of DIAC.

## n) AFX2/AFX3 Data Form

EGFG - problem no. └── AFX2  
                          └── AFX3

This form contains forward angular fluxes (AFX2) and adjoint angular fluxes (AFX3) generated by the two-dimensional  $S_N$ -transport code ESPRIT.  
 The identifications for the node names are as follows :

- level 1 : EGFG is the same as the SFX2/SFX3 data form.
- level 2 : problem no. is the same as the SFX2/SFX3 data form.
- level 3 : AFX2 shows forward angular fluxes of ESPRIT.  
 AFX3 means adjoint angular fluxes of ESPRIT.

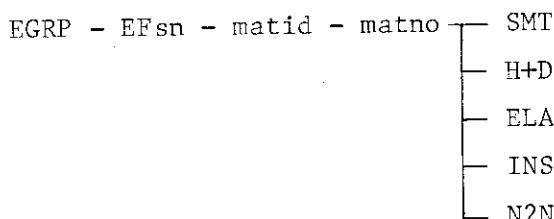
## o) RESD Data Form

EGRP - RESD - detector name

This form contains response functions to calculate reaction rates. The identifications for the node names are as follows :

- level 1 : EGRP shows the energy group structure.
- level 2 : RESD means the response data.
- level 3 : detector name indicates identification names of the detector response functions.

## p) EFsn Data Form



This form contains the effective microscopic group cross section generated by a few-group collapsing code FDEM. The identifications for the node names are as follows :

level 1 : EGRP means the energy group structure. The energy group structure is stored in the node.

level 2 : EFsn shows the effective microscopic cross section. The sn shows the number of angular meshes.

level 3 : matid shows the material identification name.

level 4 : matno shows the nuclide identification numbers contained in the material. The names are ordinarily the same as the material numbers in ENDF/B file.

level 5 : SMT, H+D, ELA, INS and N2N show reaction types of smooth cross section, energy deposition and atomic displacement, elastic scattering matrix, inelastic scattering matrix and (n,2n) scattering matrix, respectively.

## g) BREM Data Form

EGRP - FXsn - matid - BREM

This form contains the Bremsstrahlung data. The secondary gamma-ray production data generated by the Bremsstrahlung effect are stored in the node. The identifications for the node names are as follows :

level 1 : EGRP shows the energy group structure. The energy group structure is contained in the node.

level 2 : FXsn shows the macroscopic cross section.

level 3 : matid shows the material identification name.

level 4 : BREM shows the Bremsstrahlung data. The secondary gamma-ray production data by the Bremsstrahlung effect.

The record formats of data generated by the RADHEAT-V4 code system are classified according to the data forms noted above and described in the following Sections. The user information of 5 words is stored by using PSET subroutine and read by using PFIND subroutine in the DATA-POOL access package, respectively. The data in the Data Section are stored by using PRITE - PRITE4 subroutines and read by using PREAD - PREAD4 subroutines. In the following description, an "information" means the user information in the Directory Section and a "data" indicates the data in the Data Section. The node name with a capital letter shows the fixed name, and a small letter means that the name changes for each data.

### 6.1 ULTX Data Form

level 1 node : ULTX  
 information NGRP, 0, 0, 0, 0  
 data PREAD1 (N, NCOM, NGRP+1, FEGRP)

level 2 node : matno  
 information MATNO, MTMAX, NTMP, NSIG, LFI  
 data PREAD3 (N, NCOM, MTMAX, MTYPE, NTMP, TMP, NSIG, SIGO)

level 3 node : TMPi  
 information TMP, 0, 0, 0, 0  
 data PREAD (N, NCOM)

level 4 node : SIGj  
 information SIG, MTMAX2, 0, 0, 0  
 data PREAD2 (N, NCOM, MTMAX2, MTYPE2, NGRP, W)  
 DO 1 I = 1, MTMAX2  
 1 PREAD2 (N, NCOM, 5, NDATA, M, GCS)

where NGRP : number of the ultra-fine energy groups,

N : logical unit number of DATA-POOL,

NCOM : comment of the node (20 words),

MATNO : material identification number,

MTMAX : number of reactions,

NTMP : number of temperatures,  
 NSIG : length of  $\sigma_0$  table,  
 LFI : fission flag (0: non fission, 1: fission),  
 MTMAX2 : number of reactions for each  $\sigma_0$  value,  
 FEGRP : energy group boundaries (eV),  
 MTYPE : reaction identification numbers,  
 TMP : temperatures,  
 SIG0 :  $\sigma_0$  values,  
 MTYPE2 : reaction identification numbers,  
 W : weighting spectrum,  
 NDATA : MTYPE(i), C1, C2, NLOW, NUP,  
 M : NUP-NLOW+1,  
 GCS : ultra-fine group cross section from the group NLOW to NUP.

## 6.2 SMT Data Form

```

level 1 node : EGRP
information ING, IGG, 0, 0, 0
data          PREAD1(N, NCOM, ING+1, GNG)           (IGG=0)
              PREAD2(N, NCOM, ING+1, GNG, IGG+1, GGG) (IGG#0)

level 2 node : INFX
information 0, 0, 0, 0, 0
data          PREAD (N, NCOM)

level 3 node : matno
information MATNO, 0, 0, 0, 0
data          PREAD (N, NCOM)

level 4 node : SMT
information 0, 0, 0, 0, 0
data          PREAD3(N, NCOM, M, MT, 1, TMP, 1, SIG0)
              PREAD1(N, NCOM, MM, SMT)
where    ING : number of neutron energy groups,
        IGG : number of gamma-ray energy groups,
        N : logical unit number of DATA-POOL,
        NCOM : comment of the node (20 words),
        MT : reaction identification numbers,
  
```

TMP : temperature,  
 SIGO :  $\sigma_0$  value,  
 M : number of reactions (10),  
 MM : ING×M,  
 SMT : smooth cross section.

### 6.3 FTB Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : INFX  
 information same as the SMT data form  
 data ditto.

level 3 node : matno  
 information same as the SMT data form  
 data ditto.

level 4 node : FTB  
 information 0, 0, 0, 0, 0  
 data PREAD3(N, NCOM, M, MT, NTMP, TMP, NSIG, SIGO)  
     DO 1 I=1, NTMP  
     1 PREAD4(N, NCOM, LEN, SFT, LEN, SFE, LEN, SFF, LEN, SFC)  
 where M : number of reactions (4),  
       MT : reaction identification numbers,  
       NTMP : number of temperatures,  
       NSIG : number of  $\sigma_0$  values,  
       N : logical unit number of DATA-POOL,  
       NCOM : comment of the node (20 words),  
       LEN : NSIG×ING,  
       TMP : temperatures,  
       SIGO :  $\sigma_0$  values,  
       SFT : self-shielding factor for total reaction,  
       SFE : self-shielding factor for elastic reaction,  
       SFF : self-shielding factor for fission reaction,  
       SFC : self-shielding factor for capture reaction.

## 6.4 ELA Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : INFX  
 information same as the SMT data form  
 data ditto.

level 3 node : matno  
 information same as the SMT data form  
 data ditto.

level 4 node : ELA  
 information 0, 0, 0, 0, 0  
 data PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIG0)  
       DO 1 I=1, ING, 10  
       1 PREAD3(N, NCOM, ING×10, NOA, NTP, ANG, NTP, SIG)  
 where     N : logical unit number of DATA-POOL,  
           NCOM : comment of the node (20 words),  
           MT : reaction identification number (MT=2),  
           TMP : temperature,  
           SIG0 :  $\sigma_0$  value,  
           NOA : number of angular points for each energy group,  
           NTP : summation of NOA(M) values from M=1 to M=ING×10,  
           ANG : cosine of scattering angles,  
           SIG : elastic scattering cross section in the DAR form.

## 6.5 INS Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : INFX  
 information same as the SMT data form  
 data ditto.

level 3 node : matno  
 information same as the SMT data form  
 data ditto.

level 4 node : INS  
 information 0, 0, 0, 0, 0  
 data PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIG0)  
       DO 1 I=1, ING, 10  
       1 PREAD3(N, NCOM, ING×10, NOA, NTP, ANG, NTP, SIG)  
 where MT : reaction identification number (MT=4),  
       SIG : inelastic scattering cross section in the DAR form, the other  
       notations are the same as the ELA data form.

## 6.6 N2N Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : INFX  
 information same as the SMT data form  
 data ditto.

level 3 node : matno  
 information same as the SMT data form  
 data ditto.

level 4 node : N2N  
 information 0, 0, 0, 0, 0  
 data PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIG0)  
       DO 1 I=1, ING, 10  
       1 PREAD3(N, NCOM, ING×10, NOA, NTP, ANG, NTP, SIG)  
 where MT : reaction identification number (MT=16),  
       SIG : (n,2n) scattering cross section in the DAR form, the other  
       notations are the same as the ELA data form.

## 6.7 H+D Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : INFX  
 information same as the SMT data form  
 data ditto.

level 3 node : matno  
 information same as the SMT data form  
 data ditto.

level 4 node : H+D  
 information 0, 0, 0, 0, 0  
 data PREAD3(N, NCOM, M, MT, 1, TMP, 1, SIGO)  
           PREAD1(N, NCOM, MM, HD)  
 where    N : logical unit number of DATA-POOL,  
           NCOM : comment of the node (20 words),  
           M : number of reaction channels (M=13),  
           MT : reaction identification numbers,  
           TMP : temperature,  
           SIGO :  $\sigma_0$  value,  
           MM : ING×M,  
           HD : energy deposition factors and atomic displacement cross  
           sections.

## 6.8 SGRX Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : SGRX  
 information 0, 0, 0, 0, 0  
 data PREAD (N, NCOM)

```

level 3 node : matno
information MATNO, 0, 0, 0, 0
data      PREAD (N, NCOM)

level 4 node : ncode
information ITWO, ICON, KEY, NHI, NLOW
data      PREAD3(N, NCOM, LEN, X, LEN, Y, LEN1, P)
where MATNO : material identification number,
      N : logical unit number of DATA-POOL,
      NCOM : comment of the node (20 words)
      ITWO : flag of the nuclear data
              (1: ENDF/B-IV, 2: POPOP4),
      ICON : flag of the weighting procedure
              (0: constant weighting, 1: energy weighting),
      KEY : flag of the reaction
              (0: no effect, 1: inelastic excitation),
      NHI : the highest energy group for non-zero values,
      NLOW : the lowest energy group for non-zero values,
      LEN : NHI-NLOW+1,
      LEN1 : IGG×LEN,
      X : neutron interaction cross sections,
      Y : yields,
      P : probabilities ((P(i,j), i=1, IGG), j=1, LEN).

```

## 6.9 FXsn Data Form

```

level 1 node : EGRP
information same as the SMT data form
data      ditto.

level 2 node : FXsn
information IPO, 0, 0, 0, 0
data      PREAD1(N, NCOM, IPO+1, ANG)

level 3 node : matid
information MATID, IHS, IHT, IHM, NUP
data      PREAD4(N, NCOM, NMAT, MAT1, NMAT, MAT2, NMAT, ATOM, NMAT, TMP)
          DO 1 I=1, ING+IGG

```

1 PREAD2(N, NCOM, IGT1, CRX, IGT2, CRY)  
 where IPO : number of fixed angular points (IPO=sn),  
 N : logical unit number of DATA-POOL,  
 NCOM : comment of the node (20 words),  
 MATID : material identification name,  
 IHS : position of self-scattering cross section,  
 IHT : position of total cross section,  
 IHM : cross section table length,  
 NUP : table length for up-scattering,  
 NMAT : number of nuclides in the material,  
 MAT1 : nuclide identification numbers for the SMT data,  
 MAT2 : nuclide identification numbers for the FTB data,  
 ATOM : atomic number densities (n/barn·cm),  
 TMP : temperatures,  
 LGT1 : IHM,  
 LGT2 : IPO×(i+NUP),  
 CRX : effective macroscopic cross section  $\Sigma_g$ ,  
 CRY : effective macroscopic cross section  $\Sigma_{g \rightarrow g', m}$ .

In the data, CRX and CRY are defined by the following sequences,

Position	1 ----- NOACT+1 -- IHT IHT+1* IHT-NUP -- IHS ----- IHM
CRX	$\Sigma_{\text{activation}} - \Sigma_a - \bar{\nu} \Sigma_f - \Sigma_t - \Sigma_t^{\text{up}} - \Sigma_{g+NUP \rightarrow g} - \Sigma_{gg} - \Sigma_{g \rightarrow g'} - \Sigma_{1 \rightarrow g} - 0.0$

\*) omit this record when NUP=0.

where NOACT is the number of the activation cross sections consisted of the energy deposition factor and the atomic displacement cross section.

The above sequence repeats ING+IGG times.

angle No.	1	2	3	-----	IPO
1	$\Sigma_{g+NUP \rightarrow g}(\mu_1)$	$\Sigma_{g+NUP \rightarrow g}(\mu_2)$	$\Sigma_{g+NUP \rightarrow g}(\mu_3)$	-----	$\Sigma_{g+NUP \rightarrow g}(\mu_{ipo})$
2					
⋮	⋮	⋮	⋮	⋮	⋮
NUP+1	$\Sigma_{g \rightarrow g}(\mu_1)$	$\Sigma_{g \rightarrow g}(\mu_2)$	$\Sigma_{g \rightarrow g}(\mu_3)$	-----	$\Sigma_{g \rightarrow g}(\mu_{ipo})$
NUP+2	$\Sigma_{g-1 \rightarrow g}(\mu_1)$	$\Sigma_{g-1 \rightarrow g}(\mu_2)$	$\Sigma_{g-1 \rightarrow g}(\mu_3)$	-----	$\Sigma_{g-1 \rightarrow g}(\mu_{ipo})$
⋮	⋮	⋮	⋮	⋮	⋮
NUP+g	$\Sigma_{1 \rightarrow g}(\mu_1)$	$\Sigma_{1 \rightarrow g}(\mu_2)$	$\Sigma_{1 \rightarrow g}(\mu_3)$	-----	$\Sigma_{1 \rightarrow g}(\mu_{ipo})$

where CRY data are stored by starting at top left corner, sweeping from left to right, then from top to bottom. The sequence repeats ING+IGG times.

#### 6.10 SELF Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : SELF

information 0, 0, 0, 0, 0

data PREAD(N, NCOM)

level 3 node : matid

information MATID, NMAT, MTMAX, 0, 0

data PREAD4(N, NCOM, NMAT, MAT1, NMAT, MAT2, NMAT, ATOM, NMAT, TMP)

level 4 node : matno

information MATNO, 0, 0, 0, 0

data PREAD4(N, NCOM, ING, FTM, ING, FEM, ING, FFM, ING, FCM)

where N : logical unit number of DATA-POOL,

NCOM : comment of the node (20 words),

MATID : material identification number,

NMAT : number of nuclides in the material,

MTMAX : number of reactions (4),

MAT1 : nuclide identification number of the SMT data,

MAT2 : nuclide identification number of the FTB data,  
 ATOM : atomic number densities (n/barn·cm),  
 TMP : temperatures,  
 FTM : self-shielding factor for total cross section,  
 FEM : self-shielding factor for elastic cross section,  
 FFM : self-shielding factor for fission cross section,  
 FCM : self-shielding factor for capture cross section.

## 6.11 SFX0/SFX1 Data Form

level 1 node : EGRP  
 information same as the SMT data form  
 data ditto.

level 2 node : id. name  
 information IGE, IM, JM, IZM, MM  
 data PREAD4(N, NCOM, IM+1, R, JM+1, Z, IM, MA, IZM, MZ)  
 where IGE : identification for geometrical configuration,  

1 - slab	}	one-dimensional configuration,
2 - cylinder		
3 - sphere		
4 - (X-Y)	}	two-dimensional configuration,
5 - (R-Z)		
6 - (R-θ)		

 IM : number of interval meshes for X or R axis,  
 JM : number of interval meshes for Y, Z or θ axis  
 (for the case of one-dimension, JM=1),  
 IZM : number of zones,  
 MM : number of angular quadratures,  
 N : logical unit number of DATA-POOL,  
 NCOM : comment of the node (20 words),  
 R : spatial interval meshes for X or R axis (cm),  
 Z : spatial interval meshes for Y, Z or θ axis (cm),  
 MA : zone numbers by interval,  
 MZ : material numbers by interval.

level 3 node : SFX0/SFX1

SFX0 shows forward scalar flux and SFX1 means adjoint scalar flux for one-dimensional configuration.

information ING, IGG, ITH, 0, 0  
 data PREAD1(N, NCOM, IM×IGM, FLX)  
 where ITH : solution indicator (0=forward, 1=adjoint)  
 FLX : scalar fluxes.

### 6.12 SFX2/SFX3 Data Form

level 1 node : EGRP

information same as the SFX0/SFX1 data form,  
 data ditto.

level 2 node : id. name

information same as the SFX0/SFX1 data form,  
 data PREAD4(N, NCOM, IM+1, R, JM+1, Z, IM×JM, MA, IZM, MZ)  
 where notations are the same as those for SFX0/SFX1 data form.

level 3 node : SFX2/SFX3

SFX2 shows forward scalar flux and SFX3 means adjoint scalar flux for two-dimensional configuration.

information same as the SFX0/SFX1 data form  
 data D0 10 I=1, IGM  
 10 PREAD1(N, NCOM, IM×JM, FLX)  
 where notations are the same as those for SFX0/SFX1 data form.

### 6.13 AFX0/AFX1 Data Form

level 1 node : EGRP

information same as the SMT data form,  
 data ditto.

level 2 node : id. name

information same as the SFX0/SFX1 data form,  
 data ditto.

level 3 node : AFX0/AFX1

AFX0 shows forward angular flux and AFX1 means adjoint angular flux for two-dimensional configuration.

```
information ING, IGG, ITH, MM, IPMESH
data      PREAD3(N, NCOM, MM, W, MM, DSN, IPMESH, NOANLL)
          DO 1 I=1, IGM
1      PREAD1(N, NCOM, MM*IPMESH, AFX)
```

where    ITH : solution indicator (0=forward, 1=adjoint),  
           MM : number of angular quadratures,  
           IPMESH : number of spatial intervals,  
           W : angular quadrature weights,  
           DSN : angular quadrature cosines,  
           NOANLL : spatial interval numbers,  
           AFX : angular fluxes.

#### 6.14 AFX2/AFX3 Data Form

level 1 node : EGRP  
 information same as the SMT data form.  
 data        ditto.

level 2 node : id. name  
 information same as the SFX2/SFX3 data form,  
 data        ditto.

level 3 node : AFX2/AFX3  
 AFX2 shows forward angular flux and AFX3 means adjoint angular flux for two-dimensional configuration.  
 information same as the AFX0/AFX1 data form  
 data PREAD4(N, NCOM, MM, W, MM, AMU, MM, ETA, IPMESH, NOANLL)
 DO 1 I=1, IGM
 DO 1 J=1, IPMESH
1 PREAD1(N, NCOM, MM\*IM, AFX)  
 where    W : angular quadrature weights,  
 AMU : angular quadrature cosines for  $\mu$ ,  
 ETA : angular quadrature cosines for  $\eta$ ,  
 IPMESH : number of spatial interval meshes for Y, Z or  $\theta$  axis,  
 AFX : angular fluxes.

## 6.15 RESD Data Form

level 1 node : EGRP

information same as the SMT data form,  
data ditto.

level 2 node: RES

information 0, 0, 0, 0, 0  
data PREAD(N, NCOM)

level 3 node : matid

information IGM, IFLAG, 0, 0, 0  
data PREAD1(N, NCOM, IGM, RD)

where IGM : number of energy groups,  
IFLAG : detector identification (1 for neutron, 2 for gamma-ray),  
RD : detector response function.

## 6.16 EFsn Data Form

level 1 node : EGRP

information same as the SMT data form,  
data ditto.

level 2 node : EFsn

information IPN, 0, 0, 0, 0  
data PREAD1(N, NCOM, IPN+1, ANG)  
where IPN : number of angular meshes (IPN=sn),  
ANG : angular meshes.

level 3 node : matid

information MATID, NMAT, 0, 0, 0  
data PREAD4(N, NCOM, NMAT, MAT1, NMAT, MAT2, NMAT, ATOM, NMAT, TMP)  
where notations are the same as those for the SELF data form.

level 4 node : matno

information MATNO, 0, 0, 0, 0  
data PREAD (N, NCOM)

```

level 5 node : SMT
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, 10, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, 10×INGF, CRXF)

level 5 node : H+D
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, 13, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, 13×INGF, CRXF)

level 5 node : ELA
information NUPF, 0, 0, 0, 0
data        PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)

level 5 node : INS
information NUPF, 0, 0, 0, 0
data        PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)

level 5 node : N2N
information NUPF, 0, 0, 0, 0
data        PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)
where   MT   : reaction type identification numbers,
        TMP   : temperature (K),
        SIGO : background cross section,
        INGF : number of energy groups,
        IPN  : number of angular meshes,
        NUPF : number of up-scattering groups,
        CRXF : effective microscopic cross sections
                (the form is the same as that of CRX in the FXsn data form),
        CRYF : effective microscopic scattering matrix
                (the form is the same as that of CRY in the FXsn data form).

```

## 6.17 BREM Data Form

level 1 node : EGRP

information same as the FXsn data form,  
data ditto.

level 2 node : FXsn  
information same as the FXsn data form,  
data ditto.

level 3 node : matid  
information same as the FXsn data form,  
data ditto.

level 4 node : BREM  
information 0, 0, 0, 0, 0  
data PREAD1(N, NCOM, IGG×IGG, BR)  
where IGG : number of gamma-ray energy groups,  
BR(k,i) : Bremsstrahlung data from group i to k.

#### 6.18 Sample Program to Retrieve Data in DATA-POOL

To retrieve the data described in the previous Sections, the user may produces the computer program with the access package of the DATA-POOL. The sample program to obtain cross sections from the ULTX data form in the DATA-POOL is shown below. In the program, the next data are stored in one-dimensional arrays.

Item	array
The ultra-fine energy group boundaries	: FEGRP
The identification number of reactions	: MTYPE, MTYPE2
The temperature	: TMP
The background cross sections	: SIG0
The weighting spectrum	: W
The ultra-fine group cross sections of each reaction	: CROS

The sample program is written by FORTRAN language as follows :

```

CHARACTER JCONTR(40),      NODE(4),      NCOM(20)
DIMENSION FEGRP(NGRP+1),   MTYPE(MTMAX),  TMP(NTMP),  INF(12)
DIMENSION SIGO(NSIG),     NDATA(NOARY),  MTYPE2(MTMAX2)
DIMENSION W(NGRP),        NDATA(5),     GCS(NGRP),  CROS(NGRP, MTMAX2)

```

CC - DATA-POOL OPEN

```

N=91
CALL POPEN(N, JCONTR)
NODE(1)='ULTX'
NODE(2)='1192'
NODE(3)='TMP1'
NODE(4)='SIG1'
NTH=1

```

CC - READ DATA OF THE FIRST LEVEL NODE

```

CALL PFIND(N, NODE, NTH, INF, L)
IF(L.NE.0) GO TO 9000
NGRP=INF(8)
CALL PREAD1(N, NCOM, I, FEGRP)

```

CC - READ DATA OF THE SECOND LEVEL NODE

```

NTH=2
CALL PFIND(N, NODE, NTH, INF, L)
IF(L.NE.0) GO TO 9000
MATNO=INF(8)
MTMAX=INF(9)
NTMP=INF(10)
NSIG=INF(11)
LFI=INF(12)
CALL PREAD3(N, NCOM, MTMAX, MTYPE, NTMP, TMP, NSIG, SIGO)

```

CC - READ DATA OF THE THIRD LEVEL NODE

```

NTH=3
CALL PFIND(N, NODE, NTH, INF, L)
IF(L.NE.0) GO TO 9000
TMP=INF(8)
CALL PREAD(N, NAME1, NAME2, NCOM, NASBD, NOARY, NDATA)

```

CC - READ DATA OF THE FOURTH LEVEL NODE

```
NTH=4
CALL PFIND(N, NODE, NTH, INF, L)
IF(L.NE.0) GO TO 9000
SIG=INF(8)
MTMAX2=INF(9)
CALL PREAD2(N, NCOM, MTMAX2, MTYPE2, NGRP, W)
DO 1 I=1, MTMAX2
CALL PREAD2(N, NCOM, II, NDATA, M, GCS)
NLOW=NDATA(4)
NUP=NDATA(5)
DO 3 J=1, NGRP
3 CROS(J, I)=0.0
K=0
DO 2 J=NLOW, NUP
K=K+1
2 CROS(J, I)=GCS(K)
1 CONTINUE
-
-
9000 WRITE(6,9001) L
9001 FORMAT(5X, 'THE SPECIFIED NODE IS NOT FOUND CODE=', I5)
STOP
```

The data defined by the other forms can be read by the same manner.

ULTX — MATNO — TMP<sub>N</sub> — SIG<sub>N</sub>

EGRP — SGRX — MATNO — NCODE

SMT

FTB

INFX — MATNO — ELA

INS

N2N

H+D

FXsn — MATID — BREM

SELF — MATID — MATNO

PB.ID — SFX0

SFX1

SFX2

SFX3

AFX0

AFX1

AFX2

AFX3

RESD — DETID

SMT

ELA

INS

N2N

EGFG — EFsn — MATID — MATNO

H+D

Fig. 6.1 Node tree structure adopted in the RADHEAT-V4 code system

## 7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

## Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPool code. The comparison between the DATA-POOL access package and the DATAPool code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

## References

- 1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).
- 2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).
- 3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).
- 4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

## 7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

### Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPool code. The comparison between the DATA-POOL access package and the DATAPool code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

### References

- 1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).
- 2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).
- 3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).
- 4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

## 7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

### Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPool code. The comparison between the DATA-POOL access package and the DATAPool code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

### References

- 1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).
- 2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).
- 3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).
- 4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

## Appendix A Error Message of DATA-POOL

Various error messages are prepared to avoid abnormal operations.

The messages printed by the access subroutines are described below.

Note that the adequate message may not be displayed when logical errors exist in the user's program or the DATA-POOL is destroyed. The possible causes when the messages are printed are also described.

error message	subroutine	comment
<b>**ERROR CATLST**</b>		
THE NUMBER OF DIRECTORY IS LESS THAN ZERO	CATLST	NAL or NA2 in the Control Section may be destroyed.
<b>**DCLEAR ERROR**</b>		
SUBDIRECTORY CAN NOT BE SEARCHED	DCLEAR	Node name is not found in the sub-directory. The specified node name may be mistake.
ERROR-STOP (DATA POOL ALLOCATION ERROR)	NODEER	Error routine is called. Error is occurred at the node name of the level allocated to the logical unit number.
LOGICAL UNIT NUMBER		
NODE LEVEL		
NODE NAME		
THIS DIRECTORY ('XXXX') DOES NOT HAVE SUBDIRECTORY	NODEER	The specified node name may be mistake or the Directory Section may be destroyed.
ALL NODE NAMES IN LEVEL X YYYY ZZZZ ....	NODEER	The specified node name is mistake. All node names stored in the level are printed.
LOWER NODE NAME NOT EXISTS	NODEER	The address in the sub-directory may be destroyed.
THIS NODE NAME EXISTS IN DATA-POOL	NODEER	The node name is already exist. The Directory Section is destroyed
++COMPUTER MULFUNCTION++		by abnormal operations.

error message	subroutine	comment
**PDELT ERROR** THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PDELT	The level of the node is equal or less than zero. The specified node level may be mistake.
**PDGET ERROR** THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PDGET	ditto.
THE DIRECTORY OF THE NODE ('XXXX') DOES NOT HAVE SUB-DIRECTORY	PDGET	The lower nodes for the specified node name are not exist. A logical mistake may be contained in the user's program.
THE SUB-DIRECTORY OF('XXXX') PDGET IS NOT FOUND IN THE DIRECTORY OF ('YYYY')	PDGET	The Directory indicates the existence of the sub-directory, however the node name 'XXXX' is not found. The Directory Section may be destroyed.
THE DIRECTORY OF THE NODE ('XXXX') CAN NOT BE FOUND THE NODE NAME OF THE DIRECTORY IS ('YYYY')	PDGET	The Directory indicates the existence of the sub-directory, however the node name 'XXXX' is not agreed with the name in the sub-directory. The Directory Section may be destroyed.
**PFIND ERROR** THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PFIND	The level for the node name is equal or less than zero. The error may be occurred by a logical mistake in the user's program.
DIRECTORY DOES NOT HAVE SUB-DIRECTORY THE NODE NAME OF THE DIRECTORY IS ('XXXX')	PFIND	The error is caused by the destruction of the Directory Section.

error message	subroutine	comment
THE ADDRESS OF THE DIRECTORY IS. LE. O THE NODENAME OF THE DIRECTORY IS ('XXXX')	PFIND	The error is caused by the destruction of the Directory Section.
NODE NAME INPUT ERROR	PFIND	The specified node name is not found.
**PINIT ERROR** RECORD LENGTH IS TOO LONG. LENGTH='XXXX' MAX LENGTH='YYYY'	PINIT	A physical record length is larger than LBUFFR. The error can be erased by modifications of the access subroutines.
THE NUMBER OF DATA RECORD IS TOO LARGE UNIT='XXX' TOTAL RECORD NO='YYYYY'	PINIT	The number of records is larger than 50,000. The error can be erased by modifications of the access subroutines.
**POOPEN ERROR** DDNAME=XXXXXXXX IS NOT ALLOCATED	POOPEN	The data-set name is not found or the DCB information is mismatched.
**PRITE ERROR** DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN	PRITE	Data can not be written because the writing records will exceed the allowable limit.
**PRITE1 ERROR** THE SIZE OF DATA IS LESS OR EQUAL 0	PRITE1 { PRITE4	The size of data to be written is equal or less than zero. The error may be caused by a logical mistake in the user's program.
**PWSTAT ABORT** WRITE FLAG IS ALREADY ON UNIT='XX'	PWSTAT	The exclusive access control is already operated. The latest access job may be terminated abnormally.

error message	subroutine	comment
**PWSTAT ERROR** CONTROL SECTION READ ERROR UNIT='XX'	PWSTAT	An error is occurred when the Control Section is read.
**WRTCHK ERROR** SUBDIRECTORY CAN NOT BE FOUND	WRTCHK	The sub-directory to be update can not be found. DATA-POOL may be destroyed.
DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN	PRITE1 ? PRITE4	Data can not be written because the writing records will exceed the allowable limit. DATA-POOL is full.
**PSET ERROR** THE LEVEL OF THE NODE IS. LE. 0	PSET	The level of node name to be set is equal or less than zero. The error may be caused by a logical mistake in the user's program.
THE NUMBER OF SUB-DIRECTORY PSET OF THE SAME LEVEL IS TOO LARGE NODE NAME CAN NOT BE WRITTEN		The number of nodes for the same level exceeds the allowable limit. The error can be erased by modifications of access subroutines.
DIRECTORY AREA CAN NOT BE OBTAINED	PSET	The Directory Section is full. Large directory section is needed to store the node structure.
**WARNING IN PSKIP** SKIPPED TO NEXT DATA SET (NODE NAME='XXXX')	PSKIP	The skipped record is the next data with the different node name.
**ERROR IN PSKIP** DATA SET ADDRESS WAS OVERFLOWED (IX='XXXXXXXX')		The skipped record exceeds the allowable record address.

## Appendix B Program List of DATA-POOL Access Package

## DIRECTORY LIST OF J3679.DPOOL2.FORT77

***** MEMBER NAME *****	***** PAGE NO. *****	***** NO. OF CARDS *****
(NO.=001) CATLST	0001	150
(NO.=002) DATDLT	0003	49
(NO.=003) DCLEAR	0004	155
(NO.=004) DIRDLT	0007	27
(NO.=005) NODEER	0008	52
(NO.=006) PAGET	0009	6
(NO.=007) PASTO	0009	6
(NO.=008) PDELT	0009	133
(NO.=009) POGET	0012	97
(NO.=010) PFIKD	0014	95
(NO.=011) PINIT	0015	142
(NO.=012) POPEN	0018	55
(NO.=013) PREAD	0019	16
(NO.=014) PREAD1	0020	45
(NO.=015) PREAD2	0020	29
(NO.=016) PREAD3	0021	32
(NO.=017) PREAD4	0022	32
(NO.=018) PRITE	0022	51
(NO.=019) PRITE1	0024	58
(NO.=020) PRITE2	0025	60
(NO.=021) PRITE3	0026	61
(NO.=022) PRITE4	0027	64
(NO.=023) PSET	0028	212
(NO.=024) PSKIP	0032	36
(NO.=025) PWEND	0033	26
(NO.=026) PWSTAT	0034	35
(NO.=027) SETHSG	0034	10
(NO.=028) SUBDLT	0035	15
(NO.=029) WRTCHK	0035	47

-----  
1796 CARDS

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

```
*****
** CATLST **
*****
```

```

      SUBROUTINE CATLST(NUNIT)          00000100
C 1. FUNCTION                      00000200
C (1) PRINT OUT DIRECTORY TABLE    00000300
C                                     00000400
C---- 0. DECLARATION               00000500
C                                     00000600
C
C     COMMON /DPCONT/ LCONTR,NCONTR,I CONTR(40,99),IX,NSUBDS,NDSTAT
C     COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
C                      * NADWN,NADAT,NDASET,NOATE(2),NINFORM(5),NUTOLD,
C                      * NTHOLD,NODOLD(10,2),NA1
C
C     DIMENSION IFMAT(18),IIFMAT(2),IRFMAT(2),IAFMAT(2),INFORM(5)
C     CHARACTER I FMAT*4,IIFMAT*4,IRFMAT*4,IAFMAT*4,NCHR*1,MCHR*43,
C                 * IBLANK*4,WORD*4
C
C     DATA I FMAT(1),I FMAT(2),I FMAT(3),I FMAT(6),I FMAT(9),I FMAT(12),
C     * I FMAT(15),I FMAT(18)/*'(1H ',',',7X ',', ',' ,',',',',
C     * ',',',')' '
C     DATA IIFMAT(1),IIFMAT(2)/*' ', 114'/
C     DATA IRFMAT(1),IRFMAT(2)/*'1PE1','4.4'/
C     DATA IAFMAT(1),IAFMAT(2)/*'10X,', 'A4'/
C     DATA IBLANK// ''
C     DATA MCHR/'ABCDEFGHIJKLMNPQRSTUVWXYZ0123456789* +-=<>'/
C     DATA NOCHR/43/
C
C---- 1. INPUT AND PRINT OUT CONTROL SECTION
C
C     IHAX=2**30                     00002800
C     IWFILE = NUNIT                  00002900
C     READ(NUNIT,REC=1)      (ICONTR(LP1,IWFILE),LP1=1,LCONTR) 00003000
C     WRITE(6,7010)                   00003100
C     WRITE(6,7020)                   00003200
C     WRITE(6,7030)      (ICONTR(LP1,IWFILE),LP1=1,18) 00003300
C     WRITE(6,7040)      ICONTR(21,IWFILE) 00003400
C     WRITE(6,7050)      ICONTR(22,IWFILE) 00003500
C     WRITE(6,7060)      ICONTR(23,IWFILE) 00003600
C     WRITE(6,7062)      ICONTR(24,IWFILE) 00003700
C     WRITE(6,7064)      ICONTR(25,IWFILE) 00003800
C     WRITE(6,7070)      ICONTR(26,IWFILE) 00003900
C     WRITE(6,7080)      ICONTR(27,IWFILE) 00004000
C     WRITE(6,7090)      ICONTR(28,IWFILE) 00004100
C     WRITE(6,7100)      ICONTR(29,IWFILE) 00004200
C     WRITE(6,7110)      ICONTR(30,IWFILE) 00004300
C     WRITE(6,7120)      ICONTR(31,IWFILE) 00004400
C     MAXLOP = ICONTR(22,IWFILE)-ICONTR(21,IWFILE) 00004500
C     IF(MAXLOP.GT.0)           GO TO 2010 00004600
C     WRITE(6,7910)           GO TO 3100 00004700
C
C---- 2. READ AND OUTPUT DIRECTORY
C
```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

```

C
2010 IX = ICONTR(21,1WFILE)
      WRITE(6,7130)                                     00005100
C
DO 2900 LP1 = 1,MAXLOP
      IXADRS = IX                                     00005200
      READ(NUNIT,REC=IX) 1BUFFR(LP2),LP2=1,LRECO0)    00005300
      IX = IX + 1                                     00005400
      WRITE(6,7140)  IXADRS,1BUFFR(1)                 00005500
      WRITE(6,7150)  1BUFFR(3)                         00005600
      WRITE(6,7160)  1BUFFR(4)                         00005700
      IF(1BUFFR(4).LE.0)                               GO TO 2900 00005800
      DO 2800 LP2=1,1BUFFR(4)
          NASTAT=4+12*(LP2-1)+1                      00005900
          NALAST=NASTAT-1+7                           00006000
          WRITE(6,7170)                               00006100
C
      IF(1BUFFR(NASTAT+1).EQ.1BLANK) 1BUFFR(NASTAT+1)=0 00006200
      WRITE(WORD,'(A4)') 1BUFFR(NASTAT+1)             00006300
      IF(WORD.EQ.1BLANK) 1BUFFR(NASTAT+1)=0          00006400
      WRITE(6,7180) LP2,(1BUFFR(LP3),LP3=NASTAT,NALAST) 00006500
      NASTAT=NASTAT+7                               00006600
      NALAST=NASTAT-1+5                           00006700
      NC=0                                         00006800
      NFADRS=4                                         00006900
      DO 2810 LP3=NASTAT,NALAST
          NC=NC+1                                     00007000
          INFOMW(NC)=1BUFFR(LP3)                     00007100
C
      TEST IF CHARACTER
          WRITE(NCHR,'(A1)') INFOMW(NC)               00007200
      DO 2805 N=1,NOCHR
          IF(NCHR.EQ.MCHR(N:N))                      GO TO 2807 00007300
      CONTINUE                                         00007400
      GO TO 2806                                     00007500
C
2807
      IFMAT(NFADRS)=IAFMAT(1)                       00007600
      NFADRS=NFADRS+1                                00007700
      IFMAT(NFADRS)=IAFMAT(2)                       00007800
      GO TO 2830                                     00007900
C
2806
      IWRK=INFOMW(NC)                                00008000
      IWRK=IABS(IWRK)                                00008100
C
      TEST IF INTEGER
          IF(IWRK.GE.IMAX)                           GO TO 2820 00008200
          IFMAT(NFADRS)=IIFMAT(1)                     00008300
          NFADRS=NFADRS+1                            00008400
          IFMAT(NFADRS)=IIFMAT(2)                     00008500
          GO TO 2830                                     00008600
C
2820
      IFMAT(NFADRS)=IRFMAT(1)                       00008700
      NFADRS=NFADRS+1                                00008800
      IFMAT(NFADRS)=IRFMAT(2)                       00008900
C
2830
      NFADRS=NFADRS+2                                00009000
C
      CONTINUE                                         00009100
      WRITE(6,7175)                                 00009200
      WRITE(6,1FHAT)      (INFOMW(LP3),LP3=1,5)    00009300
C
2800  CONTINUE                                         00009400
2900  CONTINUE                                         00009500
C

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C----- FORMAT
C
 7010 FORMAT(1H1,/40X,'D I R E C T O R Y   L I S T')          00010600
 7020 FORMAT(1H0,'***** C O N T R O L   S E C T I O N   *****') 00010700
    * /1H0,5X,'COL.')
 7030 FORMAT(1H0,4X,'1-18   TITLE   : '/1H0,6X,18A4)           00010800
 7040 FORMAT(1H0,6X,'21   ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE 00010900
    *: ',I10)
 7050 FORMAT(1H0,6X,'22   HEAD ADDRESS FOR THE VACANT DIRECTORY AREA 00011100
    *: ',I10)
 7060 FORMAT(1H0,6X,'23   HEAD ADDRESS FOR THE VACANT DATA AREA 00011200
    *: ',I10)
 7062 FORMAT(1H0,6X,'24   WRITE FLAG 00011300
    *: ',I10)
 7064 FORMAT(1H0,6X,'25   READ FLAG (NOT USED) 00011400
    *: ',I10)
 7070 FORMAT(1H0,6X,'26   LENGTH OF THE ONE PHYSICAL RECORD 00011500
    *: ',I10)
 7080 FORMAT(1H0,6X,'27   MAXIMUM NUMBER OF THE SAME LEVEL NODE 00011600
    *: ',I10)
 7090 FORMAT(1H0,6X,'28   SIZE OF THE DIRECTORY SECTION 00011700
    *: ',I10)
 7100 FORMAT(1H0,6X,'29   SIZE OF THE DATA SECTION 00011800
    *: ',I10)
 7110 FORMAT(1H0,6X,'30   REAL NUMBER OF THE DIRECTORY RECORDS 00011900
    *: ',I10)
 7120 FORMAT(1H0,6X,'31   REAL NUMBER OF THE DATA SET RECORDS 00012000
    *: ',I10)
 7910 FORMAT(1H0,5X,'***** ERROR CATLST *****',               00012100
    */15X,'THE NUMBER OF DIRECTORY IS LESS THAN ZERO')
 7130 FORMAT(1H0,'***** D I R E C T O R Y   S E C T I O N   ', 00012200
    *9('*'))
 7140 FORMAT(1H0,2X,'*** INDEX = ',I4,' ***/1H0,9X,'NODE NAME = ',A4) 00012300
 7150 FORMAT(1H0,9X,'ADDRESS FOR THE UPPER NODE DIRECTORY = ',I8) 00012400
 7160 FORMAT(1H0,9X,'NUMBER OF THE LOWER NODE = ',I8) 00012500
 7170 FORMAT(1H0,4X,'NO.',4X,'NODE',3X,'NRECS',5X,'NADWN',5X,'NADAT', 00012600
    * 4X,'NDASET',5X,'DATE')
 7175 FORMAT(1H0,14X,'INFOH(1)',6X,'INFOH(2)',6X,               00012700
    * 'INFOH(3)',6X,'INFOH(4)',6X,'INFOH(5)')
 7180 FORMAT(1H ,3X,14,4X,A4,AX,I4,3(2X,18),2X,2A4)           00012800
C
      RETURN
 3100 STOP
END

```

```

*****
** DATDLT  **
*****

```

```

      SUBROUTINE DATDLT(NUNIT,NSDOLD)
C
C----- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NSTAT

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,      00000060
*          NADWN,NADAT,NASET,NDATE(2),NINFOR(5),NUTOLD,      00000070
*          NTHOLD,NODOLD(10,2),NA1      00000080
CHARACTER*4 SLASH      00000081
DATA SLASH// ////      00000082
C      00000090
DIMENSION NSDOLD(12)      00000100
C      00000110
C--- DELETE DATA      00000120
C      00000130
IF(NSDOLD(4).LE.0)      GO TO 2000      00000140
IWFIL = NUNIT      00000150
IX = NSDOLD(4)      00000160
NDTTL = 0      00000170
C      00000180
IF(NSDOLD(5).LE.0)      GO TO 2000      00000190
DO 1010 LP1=1,NSDOLD(5)
  READ(NUNIT,REC=IX) (IBUFFR(LP2),LP2=1,LRECOD)      00000200
C      IBUFFR(1) = '////'      00000210
  READ(SLASH,'(A4)') IBUFFR(1)      00000221
C      IX = IX-1      00000230
  WRITE(NUNIT,REC=IX) (IBUFFR(LP2),LP2=1,LRECOD)      00000240
  NWSIZE=0      00000250
  IF(IBUFFR(25).LE.0)      GO TO 1210      00000260
    DO 1110 LP2=1,IBUFFR(25)
      IWRK=25+LP2      00000270
      NWSIZE=NWSIZE+IBUFFR(IWRK)      00000280
    CONTINUE      00000290
1110    CONTINUE      00000300
1210    CONTINUE      00000310
  NWSIZE=NWSIZE+25+IBUFFR(25)      00000320
  MAXLOP=NWSIZE/LRECOD      00000330
  IWRK=MOD(NWSIZE,LRECOD)      00000340
C      00000350
  IF(IWRK.EQ.0)      GO TO 1220      00000360
    MAXLOP=MAXLOP+1      00000370
C1220  IX=IX-1+MAXLOP      00000380
1220  IX=IX+MAXLOP      00000381
  NDTTL = NDTTL+MAXLOP      00000390
C      00000400
1010  CONTINUE      00000410
  ICONTR(31,IWFIL)=ICONTR(31,IWFIL)-NDTTL      00000420
C      00000430
2000  RETURN      00000440
END      00000450

*****
** DCLEAR **
*****


SUBROUTINE DCLEAR      00000010
C      00000020
C--- DECLARATION      00000030
C      00000040
COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00000050
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,      00000060

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

*          NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD,      00000070
*          NTHOLD,NODOLD(10,2),NA1                           00000080
COMMON /DPWCHK/ NRPEMT,NRWRDN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3), 00000090
*          ISDBEF(12)                                       00000100
COMMON /DPDELT/ IBUFF2(1000)                                00000110
CHARACTER*4 SLASH                                         00000111
DATA SLASH'/////'                                         00000112
C                                                       00000120
C--- PUT BACK WRITE STARTING POINT(DATA INDEX) AND      00000130
C RESTORE THE COUNTER OF DATA RECORD                      00000140
C                                                       00000150
*          WRITE(6,7810) NUTWTN,NTHWTN                      00000160
*          WRITE(6,7820) (NODWTN(LP1,1),LP1=1,NTHWTN)       00000170
*          WRITE(6,7830) (NODWTN(LP1,2),LP1=1,NTHWTN)       00000180
*          WRITE(6,7830) (NODWTN(LP1,3),LP1=1,NTHWTN)       00000190
*          WRITE(6,7850) NIXOLD,NRWRDN                      00000200
*7810 FORMAT(1H ,***** DCLEAR * NUTWTN = ',14,'* NTHWTN = ',14) 00000210
*7820 FORMAT(1H ,***** DCLEAR * NODWTN = ',','*,10(A4,'*)) 00000220
*7830 FORMAT(1H ,***** DCLEAR * NODWTN = ',','*,10(A4,'*)) 00000230
*7850 FORMAT(1H ,***** DCLEAR * NIXOLD = ',19,'* NRWRDN = ',19) 00000240
  IFILE = NUTWTN                                         00000250
  ICONTR(23,IFILE) = NIXOLD                            00000260
  ICONTR(31,IFILE) = ICONTR(31,IFILE)-NRWRDN           00000270
C                                                       00000280
  NCOUNT = 0                                           00000290
  IX = ICONTR(21,IFILE)                                00000300
C                                                       00000310
C--- DELETE DIRECTORY IF IT WAS MADE JUST PRIOR TO THIS PROCESS 00000320
C                                                       00000330
1000 NCOUNT = NCOUNT+1                                 00000340
  READ(NUTWTN,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD) 00000350
  IF(NCOUNT.LE.1)                                     GO TO 2000 00000360
  NCOPRE=NCOUNT-1                                    00000370
  IF(NODWTN(NCOPRE,3).NE.0)                          GO TO 2000 00000380
C   IBUFFR(1) = '////'                                00000390
  READ(SLASH,'(A4)') IBUFFR(1)                         00000391
C   IX = IX-1                                         00000400
  WRITE(NUTWTN,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD) 00000410
  ICONTR(30,IFILE) = ICONTR(30,IFILE)-1               00000420
C*** SEARCH SUBDIRECTORY                               00000430
  2000 IANDNM = 5                                     00000440
    DO 2010 LP1=1,IBUFFR(4)                            00000450
      IORDER = LP1                                      00000460
      IF(NODWTN(NCOUNT,1).EQ.IBUFFR(IANDNM))        GO TO 2100 00000470
      IANDNM = IANDNM+12                                00000480
  2010 CONTINUE                                         00000490
    WRITE(6,7910)                                     GO TO 5100 00000500
C*** RESET SUBDIRECTORY                             00000510
  2100 IF(NCOUNT.GE.NTHWTN)                          GO TO 3000 00000520
    IF(NODWTN(NCOUNT,2).EQ.0)                          GO TO 3100 00000530
    IF(NODWTN(NCOUNT,2).NE.3)                          GO TO 2200 00000540
      IWRK=IANDNM+2                                    00000550
      IADIRC=IBUFFR(IWRK)                            00000560
      IBUFFR(IWRK) = 0                                00000570
      00000580

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C      IX = IX-1                               00000590
      WRITE(NUTWTN,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD)
                                         GO TO 2300 00000600
                                         00000610
C**** CASE DIRECTORY IS NOT CHANGED 00000620
 2200 IWRK = IANDNM+2                      00000630
      IADIRC = IBUFFR(IWRK)                   00000640
C**** SET LOWER DIRECTORY ADDRESS 00000650
 2300 IX = IADIRC                          00000660
                                         GO TO 1000 00000670
C                                         00000680
C---- SUBDIRECTORY OF THE NODE NAME THAT IS LAST OR OVERFLOWED 00000690
C                                         00000700
C
 3000 IF(NODWTN(NCOUNT,2).NE.0)           GO TO 4000 00000710
C**** NODWTN(NCOUNT,2) IS ZERO. DELETE SUBDIRECTORY 00000720
 3100 IWRK=IANDNM+2                      00000730
      IADIRC=IBUFFR(IWRK)                   00000740
      IOTHER=IBUFFR(4)-IORDER              00000750
      IF(IOTHER.LE.0)                      GO TO 3200 00000760
          IWRK = IANDNM-1                  00000770
          DO 3120 LP1=1,IOTHER             00000780
          DO 3130 LP2=1,12                00000790
              IWRK=IWRK+1                  00000800
              IWRK1=IWRK+12                00000810
              IBUFFR(IWRK) = IBUFFR(IWRK1) 00000820
 3130    CONTINUE                           00000830
 3120    CONTINUE                           00000840
C                                         00000850
 3200 IBUFFR(4)=IBUFFR(4)-1               00000860
C      IX=IX-1                           00000870
      WRITE(NUTWTN,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD)
      IF(NCOUNT.GE.NTHWTN)           GO TO 5200 00000880
          IX=IADIRC
                                         GO TO 1000 00000890
                                         00000900
                                         00000910
C**** NODWTN(NCOUNT,2) IS 4 00000920
C**** RESTORE SUBDIRECTORY AND LINK OLD DATA TO SUBDIRECTORY 00000930
 4000 IF(NODWTN(NCOUNT,2).NE.4)           GO TO 4500 00000940
      IWRK=IANDNM-1                  00000950
      DO 4010 LP1=1,12                00000960
          IWRK=IWRK+1                  00000970
          IBUFFR(IWRK)=ISDBEF(LP1)   00000980
 4010 CONTINUE                           00000990
      DO 4020 LP1=1,LRECOD            00001000
          IBUFF2(LP1)=IBUFFR(LP1)     00001010
 4020 CONTINUE                           00001020
C      IXOLD=IX-1                     00001030
      IXOLD=IX
      IF(ISDBEF(4).EQ.0)           GO TO 4400 00001040
      IX=ISDBEF(4)                 00001050
      NDTOTL = 0                   00001060
      DO 4110 LP1=1,ISDBEF(5)      00001070
          READ(NUTWTN,REC=IX)      (IBUFFR(LP2),LP2=1,LRECOD)
          IBUFFR(1)=ISDBEF(1)     00001080
          00001090
C      IX=IX-1                     00001100
      WRITE(NUTWTN,REC=IX)      (IBUFFR(LP2),LP2=1,LRECOD)
      NWSIZE=0                    00001110
                                         00001120

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY) TIME 15:00:36 PAGE 00

```

        IF(IBUFFR(25).LE.0)                      GO TO 4200      00001130
        DO 4120   LP2=1,IBUFFR(25)
                  IWRK=25+LP2
                  NWSIZE=NWSIZE+IBUFFR(IWRK)
4120      CONTINUE
4200      NWSIZE=NWSIZE+25+IBUFFR(25)
                  MAXLOP=NWSIZE/LRECOD
                  IWRK=MOD(NWSIZE,LRECOD)
                  IF(IWRK.EQ.0)                      GO TO 4300      00001170
                  MAXLOP=MAXLOP+1
C4300      IX=IX-1+MAXLOP
4300      IX=IX+MAXLOP
                  NDTOTL=NDTOTL+MAXLOP
4110      CONTINUE
                  ICNTR(31,IWFILE)=ICNTR(31,IWFILE)+NDTOTL
4400      IX=IXOLD
                  WRITE(NUTWTN,REC=IX)    (IBUFF2(LP1),LP1=1,LRECOD)
                                         GO TO 5200      00001280
                                         00001290
C**** NODWTN(NCOUNT,2) IS 3
C**** RESET ADDRESS FOR THE DIRECTORY OF THIS LOWER NODE
4500      IF(NODWTN(NCOUNT,2).NE.3)              GO TO 5200      00001310
                  IWRK=IANDNH+2
                  IADIRC=IBUFFR(IWRK)
                  IBUFFR(IWRK)=0
C         IX=IX-1                                00001340
                  WRITE(NUTWTN,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
                                         00001360
C
C---- RETURN
C
5200      RETURN
5100      ICNTR(24,IWFILE)=0
                  WRITE(NUTWTN,REC=1)    (ICNTR(LP2,IWFILE),LP2=1,LCONTR)
                                         00001420
                  STOP
C
C---- FORMAT
C
7910      FORHAT(1H ,'***** DCLEAR ERROR *****'
                  *15X,'SUBDIRECTORY CAN NOT BE SEARCHED')
                                         00001480
                  END
                                         00001490
                                         00001500

```

```

*****
** DIRDLT **
*****

```

```

        SUBROUTINE DIRDLT(NUNIT,NSDOLD)          00000010
C
C---- DECLARATION
C
        COMMON /DPWORK/ LBUFFER,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                  *      NADWN,NADAT,NASET,NDATE(2),NINFOR(5),NUTOLD,
                  *      NTHOLD,RODOLD(10,2),NA1
        COMMON /DPCNT/ LCNTR,NCONTR,ICNTR(40,99),IX,NSUBDS,NSTAT
        CHARACTER*4 SLASH
        DATA SLASH'///'
C

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

```

        DIMENSION NSDOLD(1)                      00000100
C
C---- DELETE DIRECTORY (SET '////' INTO NODE NAME OF DIRECTORY) 00000110
C
C
        IF(NSDOLD(3).LE.0)                      GO TO 2000 00000120
        IX = NSDOLD(3)                          00000130
        READ(NUNIT,REC=IX)          (IBUFFR(LP1),LP1=1,LRECOD) 00000140
C
        IBUFFR(1) ~ '////'                     00000150
        READ(SLASH,'(A4)') IBUFFR(1)           00000160
C
        IX = IX-1                            00000170
        WRITE(NUNIT,REC=IX)          (IBUFFR(LP1),LP1=1,LRECOD) 00000171
        IWFILE = NUNIT                         00000180
        I CONTR(30,IWFILE) = I CONTR(30,IWFILE)-1 00000190
C
        2000 RETURN                           00000200
        END                                  00000210
                                         00000220
                                         00000230
                                         00000240

```

\*\*\*\*\*  
\*\* NODEER \*\*  
\*\*\*\*\*

```

SUBROUTINE NODEER(NUNIT,NTH,NODE)          00000010
C
C--- DECLARATION                         00000020
C
COMMON /DPCONT/ LCONTR,NCONTR,I CONTR(40,99),IX,NSUBDS,NSTAT 00000030
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00000040
*             NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD,
*             NTHOLD,NOOLD(10,2),NA1                         00000050
C
DIMENSION NODE(1),MODE(100)                00000060
C
C--- ERROR ROUTINE ( OUTPUT NODE NAME AND DEPTH ) 00000070
C
        WRITE(6,9000) NUNIT,NTH,(NODE(N),N=1,NTH) 00000080
        NCOUNT = 0                                00000090
        IWFILE = NUNIT                           00000100
        IX = I CONTR(21,IWFILE)                  00000110
1000  NCOUNT = NCOUNT+1                    00000120
        READ(NUNIT,REC=IX) (IBUFFR(LP1),LP1=1,LRECOD) 00000130
        IF(IBUFFR(4).GT.0) GO TO 1010            00000140
        WRITE(6,9010) IBUFFR(1)                  00000150
        STOP                                     00000160
        1010 IANDNM = 5                         00000170
        NBUFA = IBUFFR(4)                        00000180
        DO 1020 LP1=1,NBUFA                     00000190
        MODE(LP1)=IBUFFR(IANDNM)                 00000200
        IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM)) GO TO 1030 00000210
        IANDNM = IANDNM+12                      00000220
1020  CONTINUE                               00000230
        WRITE(6,9020) NCOUNT,(MODE(N),N=1,NBUFA) 00000240
        STOP                                     00000250
1030  IF(NCOUNT.GE.NTH) GO TO 1040          00000260
        IWRK = IANDNM+2                         00000270
        IX = IBUFFR(IWRK)                       00000280
                                         00000290
                                         00000300
                                         00000310
                                         00000320
                                         00000330
                                         00000340

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

      IF(IX.GT.0)      GO TO 1000          00000350
      WRITE(6,9030)
      STOP
1040 WRITE(6,9040)
      STOP
9000 FORMAT('0*****' ERROR-STOP ( DATA POOL ALLOCATION ERROR ) *****' 00000400
      1   '/' LOGICAL UNIT NUMBER ----- ',15          00000410
      2   '/' NODE LEVEL      ----- ',15          00000420
      3   '/' NODE NAME       ----- ',10(1X,A4))        00000430
9001 FORMAT('0+++++' ERROR MESSAGE    +++++')        00000440
9010 FORMAT('0      THIS DIRECTORY ( ',A4,' ) DOSE NOT HAVE SUBDIREC 00000450
      1TORY')
9020 FORMAT('0      ALL NODE NAMES IN LEVEL',12,          00000460
      1     /(10(4X,A4)))
9030 FORMAT('0      LOWER NODE NAME NOT EXISTS')
9040 FORMAT('0      THIS NODE NAME EXISTS IN DATA-POOL',
      1   '/0 +++++ COMPUTER MULFUNCTION ++++++')        00000510
      END

```

```
*****
** PAGET **
*****
```

```

      SUBROUTINE PAGET(NASELF)          00000010
C   SET CURRENT ADDRESS TO NASELF          00000020
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      NASELF=IX          00000030
      RETURN          00000040
      END          00000050

```

```
*****
** PASTO **
*****
```

```

      SUBROUTINE PASTO(NASELF)          00000010
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C
      IX = NASELF          00000020
      RETURN          00000030
      END          00000040

```

```
*****
** PDELT **
*****
```

```

      SUBROUTINE PDELT(NUNIT,NODE,NTH,NRETUN)          00000100
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
      *           NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
      *           NTHOLD,NODOLD(10,2),NA1          00000200
      COMMON /DPDELT/ IBUFF2(1000)
      CHARACTER*4 CHR          00000300

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C                               00001100
      DIMENSION NODE(1),NSDOLD(12)          00001200
C                               00001300
C---- DELETE DIRECTORY AND DATA AND SUBDIRECTORY OF THE NODES WHICH IS 00001400
C      LOWER THAN NODE(NTH)                 00001500
C                               00001600
C
      IF(NTH.GT.0)                         GO TO 1010    00001700
      WRITE(6,7910)
      NRETUN=1
      GO TO 5900
1010 IFILE = NUNIT                   00002100
      READ(NUNIT,REC=1) (ICONTR(LP1,IFILE),LP1=1,LCTR) 00002200
      LRECOD=ICONTR(26,IFILE)                00002300
C      WRITE FLAG ON                      00002400
      ICONTR(24,IFILE)=1                  00002500
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IFILE),LP1=1,LCTR) 00002600
      NCOUNT = 0                          00002700
      IX = ICONTR(21,IFILE)                00002800
C
C---- SEARCH THE SUBDIRECTORY OF THE LAST NODE 00002900
C
      1100 NCOUNT = NCOUNT+1               00003000
      READ(NUNIT,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD) 00003100
      IF(IBUFFR(4).GT.0)                 GO TO 1110    00003200
      NRETUN = 1
      GO TO 5800
C
      1110 IANDNM = S                   00003800
      DO 1210 LP1=1,IBUFFR(4)           00003900
      IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))   GO TO 1300    00004000
      IANDNM = IANDNM+12
1210 CONTINUE
      NRETUN = 1
      GO TO 5800
C
      1300 IF(NCOUNT.GE.NTH)           GO TO 2000    00004400
      IWRK=IANDNM+2
      IX = IBUFFR(IWRK)
      GO TO 1100
C
C---- SAVE SUBDIRECTORY AND DELETE DIRECTORY AND DATA BY CALLING 00004500
C      SUBROUTINE SUBDLT
C
      C2000 NASUBD = IX-1              00005100
      2000 NASUBD = IX
      NOSUBD = LP1
      *      WRITE(6,7810)      NASUBD,NOSUBD        00005200
      *7810 FORMAT(1H ,5X,'---- NASUBD =',I13,      00005300
      *      */6X,'---- NOSUBD =',I13)            00005400
      **** STORE SUBDIRECTORY TO NSDOLD(12)        00005500
      DO 2100 LP1=1,12
      IWRK = IANDNM+(LP1-1)
      NSDOLD(LP1)=IBUFFR(IWRK)
2100 CONTINUE
C**** DELETE DIRECTORY AND DATA OF THE NODE(NTH) 00005600
      00005700
      00005800
      00005900
      00006000
      00006100
      00006200
      00006300
      00006400
      00006500

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

```

*      WRITE(6,7820)      NSDOLD          00006600
*7820 FORMAT(1H ,5X,'----- DIRECTORY(DEBUG) ',           00006700
*      */10X,'NODE1$ ',A4          00006800
*      */10X,'NODE2$ ',A4          00006900
*      */10X,'NADWN  = ',I13        00007000
*      */10X,'NADAT  = ',I13        00007100
*      */10X,'NDASET = ',I13        00007200
*      */10X,'DATE   = ',2A4         00007300
*      */10X,'INFORM = ',5(I13,'*') 00007400
      CALL SUBDLT(NUNIT,NSDOLD)      00007500
C**** DELETE DIRECTORY AND DATA OF THE NODE WHICH IS LOWER THAN NODE(NTH)00007600
C      )                           00007700
      NCOUNT = 0                   00007800
3000 NCOUNT=NCOUNT+1             00007900
      IX=NASUBD+NCOUNT            00008000
      IF(IX.GE.ICONTR(22,IWFILE))    GO TO 4000 00008100
      READ(NUNIT,REC=IX)           (IBUFF2(LP1),LP1=1,LRECOD) 00008200
C      IF(IBUFF2(1).NE.'//')       GO TO 3000 00008300
      WRITE(CHR,'(A4)') IBUFF2(1)     00008400
      IF(CHR.NE.'//')              GO TO 3000 00008500
      IF(IBUFF2(4).LE.0)           GO TO 3000 00008600
      NOSUBD=IBUFF2(4)             00008700
      IBUFF2(4)=0                 00008800
      WRITE(NUNIT,REC=IX)           (IBUFF2(LP1),LP1=1,LRECOD) 00008900
      IANDNM = 5                  00009000
      DO 3010 LP1=1,NOSUBD         00009100
      DO 3020 LP2=1,12              00009200
      IWRK = IANDNM+(LP2-1)         00009300
      NSDOLD(LP2) = IBUFF2(IWRK)    00009400
3020  CONTINUE                  00009500
      CALL SUBDLT(NUNIT,NSDOLD)     00009600
      IANDNM = IANDNM+12           00009700
3010  CONTINUE                  00009800
                                GO TO 3000 00009900
C
C---- DELETE SUBDIRECTORY OF THE NODE(NTH) 00010000
C
4000 CONTINUE                  00010100
      IX = NASUBD                 00010200
      READ(NUNIT,REC=IX)           (IBUFFR(LP3),LP3=1,LRECOD) 00010300
      IF(IBUFFR(4).LE.NOSUBD)      GO TO 4100 00010400
      MAXLP = IBUFFR(4)-NOSUBD    00010500
      MOVTO = 5+(NOSUBD-1)*12      00010600
      MOVFRM = 5+NOSUBD*12         00010700
      DO 4010 LP2=1,MAXLP          00010800
      DO 4020 LP3=1,12              00010900
      ITO = MOVTO+LP3-1            00011000
      IFROM = MOVFRM+LP3-1          00011100
      IBUFFR(ITO) = IBUFFR(IFROM)  00011200
4020  CONTINUE                  00011300
      MOVTO = MOVTO+12              00011400
      MOVFRM = MOVFRM+12           00011500
4010  CONTINUE                  00011600
C
4100 IBUFFR(4) = IBUFFR(4)-1     00011700
                                00011800
                                00011900
                                00012000

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C      IX = IX-1                               00012100
      WRITE(NUNIT,REC=IX)      (IBUFFR(LP3),LP3=1,LRECOD) 00012200
      NRETUN=0                                00012300
      5800 I CONTR(24,IWFILE)=0                00012400
      WRITE(NUNIT,REC=1)      (ICONTR(LP3,IWFILE),LP3=1,LCONTR) 00012500
C                                              00012600
C----- RETURN                               00012700
C                                              00012800
      5900 RETURN                             00012900
C                                              00013000
C      7910 FORMAT(1H ,10X,'***** PDELT ERROR *****'
      * /15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0') 00013100
      END                                     00013200
                                         00013300

*****
** PDGET   **
*****
```

SUBROUTINE PDGET(NUNIT,NODE,NTH,ITEM,NSDIRC) 00000100

C-----

C----- PDGET 00000200

C 1. FUNCTION 00000300

C (1) READ THE DIRECTORY OF NODE(NTH) 00000400

C 00000500

C 00000600

C 2. INPUT 00000700

C (1) ARGUMENT 00000800

C 1-1 NUNIT : INPUT FILE NO OF DIRECTORY 00000900

C 1-2 NODE(I) : NODE NAME 00001000

C 1-3 NTH : THE DEPTH OF NODE 00001100

C 00001200

C 3. OUTPUT 00001300

C (1) ARGUMENT 00001400

C 1-1 ITEM : THE NUMBER OF THE LOWER NODE OF THE DIRECTORY 00001500

C 1-2 NSDIRC(I,J) : THE ARRAY THAT SUBDIRECTORIES ARE TO BE 00001600

C STORED 00001700

C 4. RESTRICTION 00001800

C (1) NSDIRC(1,J) 00001900

C I = 12 , J,GE,ITEM 00002000

C----- 00002100

C-----

C----- DECLARATION 00002200

C----- 00002300

C----- 00002400

C----- COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NSTAT 00002500

C----- COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00002600

\* NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD, 00002700

\* NTHOLD,NODOLD(10,2),NA1 00002800

C----- 00002900

C----- DIMENSION NODE(1),NSDIRC(12,1) 00003000

C----- 00003100

C----- CHECK INPUT VARIABLES AND SET INITIAL VALUES 00003200

C----- 00003300

C----- IF(NTH.GE.1) GO TO 1010 00003400

WRITE(6,7910) 00003500

GO TO 3900 00003600

1010 NCOUNT = 0 00003700

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        IWFILE = NUNIT                      00003800
        IX = ICONTR(21,IWFILE)                00003900
        LRECOD=ICONTR(26,IWFILE)              00004000
C                                               00004100
C---- SEARCH SUB DIRECTORY OF NODE(NCOUNT) 00004200
C                                               00004300
C
1100 NCOUNT = NCOUNT+1                      00004400
      READ(NUNIT,REC=IX)      (IBUFTR(LP1),LP1=1,LRECOD) 00004500
      IANDNM = 5                           00004600
      IF(IBUFTR(4).GT.0)                   GO TO 1110    00004700
          WRITE(6,7920)      IBUFTR(1)           00004800
          GO TO 3900                     00004900
1110 IANDNM = 5                           00005000
      DO 1210 LP1=1,IBUFTR(4)             00005100
          IF(NODE(NCOUNT).EQ.IBUFTR(IANDNM))  GO TO 1300  00005200
          IANDNM = IANDNM+1                 00005300
1210 CONTINUE
      WRITE(6,7930)      NODE(NCOUNT),IBUFTR(1)       00005400
                                GO TO 3900         00005500
      00005600
1300 IWRK2 = IANDNM+2                      00005700
      IX = IBUFTR(IWRK2)                  00005800
      IF(NCOUNT.LT.NTH)                  GO TO 1100    00005900
C                                               00006000
C
C---- SET DIRECTORY(SUB-DIRECTORY) INTO NSDIRC(I,J) 00006100
C                                               00006200
C                                               00006300
C
2000 CONTINUE
      READ(NUNIT,REC=IX)      (IBUFTR(LP1),LP1=1,LRECOD) 00006400
      IF(NODE(NTH).EQ.IBUFTR(1))       GO TO 2010    00006500
          WRITE(6,7940)      NODE(NTH),IBUFTR(1)       00006600
                                GO TO 3900         00006700
      00006800
C
2010 ITEM = IBUFTR(4)                      00006900
      IF(ITEM.LE.0)                   GO TO 3800    00007000
      IANDNM = 4                     00007100
      DO 2210 LP1=1,ITEM            00007200
      DO 2220 LP2=1,12              00007300
          IANDNM = IANDNM+1          00007400
          NSDIRC(LP2,LP1) = IBUFTR(IANDNM) 00007500
      00007600
2210 CONTINUE
2220 CONTINUE
C
C---- RETURN
C
3800 RETURN
3900 STOP
C
7910 FORMAT(1H ,10X,'***** PDGET ERROR *****'      00008500
          */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0') 00008600
7920 FORMAT(1H ,10X,'***** PDGET ERROR *****'      00008700
          */15X,'THE DIRECTORY OF THE NODE ('',A4,'') DOSE NOT HAVE SUB-DIRECTORY' 00008800
          *RY')
7930 FORMAT(1H ,10X,'***** PDGET ERROR *****'      00008900
          */15X,'THE SUB-DIRECTORY OF ('',A4,'') IS NOT FOUND IN THE DIRECTORY 00009100
          *OF ('',A4,'')'               00009200

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

7940 FORMAT(1H ,10X,'***** PGET ERROR *****'          00009300
      */15X,'THE DIRECTORY OF THE NODE ('',A4,'') CAN NOT BE FOUND', 00009400
      */15X,'THE NODE NAME OF THE DIRECTORY IS ('',A4,'')'        00009500
C           END                                              00009600
                                         00009700

*****
** PFIND   **
*****

      SUBROUTINE PFIND(NUNIT,NODE,NTH,NOIRC,LLL)          00000100
C           00000200
C----- DECLARATION                                00000300
C           00000400
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000500
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
      *                  NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD,
      *                  NTHOLD,NODOLD(10,2),NA1                      00000600
      *                  00000700
      *                  00000800
C           00000900
      DIMENSION NODE(1),NOIRC(1)                         00001000
C           00001100
C----- CHECK INPUT VALUE, SET INITIAL VALUE        00001200
C           00001300
      LLL=0                                              00001400
      IF(NTH.GT.0)                                     GO TO 1010 00001500
      WRITE(6,7910)
      LLL=900
      GO TO 4200
1010 NCOUNT = 1                                      00001600
      IFILE = NUNIT
      IX = ICTR(21,IFILE)
      LRECOD=ICTR(26,IFILE)
      IF(NUNIT.NE.NUTOLD)                               GO TO 2200 00001700
      IF(NTHOLD.LE.1)                                  GO TO 2200 00001800
C           00001900
      00002000
      00002100
      00002200
      00002300
      00002400
C           00002500
C----- SEARCH SUBDIRECTORY ADDRESS BY OLD ACCESS TABLE NODOLD 00002600
C           00002700
      MAXLP1 = NTHOLD-1                               00002800
      DO 1020 LP1=1,MAXLP1
      NCOUNT = NCOUNT+1
      IF(NCOUNT.LE.NTH)                               GO TO 1030 00002900
      NCOUNT=NCOUNT-1
      GO TO 2200
1030     IF(NODOLD(NCOUNT,1).EQ.NODE(NCOUNT))      GO TO 1040 00003000
      NCOUNT=NCOUNT-1
      GO TO 2200
      00003100
      00003200
      00003300
      00003400
      00003500
      00003600
      00003700
      00003800
      00003900
C           00004000
C----- SEARCH SUBDIRECTORY ADDRESS OF NODE(NCOUNT) READING DIRECTORY OF 00004100
C           00004200
C           00004300
      2000 NCOUNT = NCOUNT+1
      2200 READ(NUNIT,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD) 00004400
                                         00004500

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        IF(1BUFFR(4).GT.0)          GO TO 2300      00004600
C       WRITE(6,7920)           1BUFFR(1)        00004700
        LLL=800                   00004800
        GO TO 4100                00004900
2300 IANDNM=5                 00005000
        DO 2310 LP1=1,IBUFFR(4)    00005100
          IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))   GO TO 2400  00005200
          IANDNM=IANDNM+12            00005300
2310 CONTINUE                  00005400
C       WRITE(6,7940)           00005500
        LLL=100+NCOUNT             00005600
        GO TO 4100                00005700
2400 NODOLD(NCOUNT,1)=IBUFFR(IANDNM) 00005800
C       NODOLD(NCOUNT,2) = IX-1  00005900
C       NODOLD(NCOUNT,2) = IX    00006000
        IF(NCOUNT.GE.NTH)         GO TO 3000      00006100
          IWRK = IANDNM+2
          IX = IBUFFR(IWRK)
          IF(IX.GT.0)              GO TO 2500      00006200
C       WRITE(6,7930)           NODE(NCOUNT)    00006300
          LLL=200+NCOUNT           00006400
          GO TO 4100                00006500
2500 CONTINUE                  00006600
        GO TO 2000                00006700
C
C---- SET SUBDIRECTORY TO NDIRC AND SET DATA ADDRESS TO IX 00006800
C
C       3000 00 3010  LP1=1,12      00006900
          NDIRC(LP1) = IBUFFR(IANDNM) 00007000
          IANDNM = IANDNM+1
3010 CONTINUE                  00007100
          NA1= IX                  00007200
          IX = NDIRC(4)
          NUTOLD = NUNIT             00007300
          NTHOLD = NTH               00007400
          00007500
          00007600
          00007700
          00007800
          00007900
          00008000
          00008100
C
          4100 RETURN                00008200
          4200 STOP                  00008300
          00008400
C
          7910 FORMAT(1H ,10X,'***** PFIND ERROR *****'
          */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0') 00008500
          7920 FORMAT(1H ,10X,'***** PFIND ERROR *****'
          */15X,'THE DIRECTORY DOES NOT HAVE SUB-DIRECTORY' 00008600
          */15X,' THE NODE NAME OF THE DIRECTORY IS (',A4,')') 00008700
          7930 FORMAT(1H ,10X,'***** PFIND ERROR *****'
          */15X,'THE ADDRESS OF THE DIRECTORY IS .LE. 0' 00008800
          */15X,'THE NODE NAME OF THE DIRECTORY IS (',A4,')') 00008900
          7940 FORMAT(1H ,10X,'***** PFIND ERROR *****'
          */15X,'NODE NAME INPUT ERROR') 00009000
          00009100
          00009200
          00009300
          00009400
          00009500
          END

*****
** PINIT   **
*****

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        SUBROUTINE PINIT (NUNIT,NDIRCT,LENGTH,ITITLE)      00000010
C                                                    00000020
C----- 0. DECLARATION                                00000030
C                                                    00000040
C     COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000050
C                                                    00000060
C     COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00000070
*          NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00000080
*          NTOLD,NODOLD(10,2),NA1 00000090
C     COMMON /DPEMSG/ IFLAG,NERNO 00000100
C                                                    00000110
C     DIMENSION ITITLE(1) 00000120
C     CHARACTER*4 BLANK 00000121
C     DATA BLANK/'    '/ 00000122
C     EXTERNAL SETMSG 00000130
C     CALL ERRSET(232,0,-1,1,SETMSG) 00000140
C                                                    00000150
C----- 1. DECLARATION OF DIRECT ACCESS FILE 00000160
C                                                    00000170
C     LRECL=4*LENGTH 00000180
C     OPEN(NUNIT,ACCESS='DIRECT',RECL=LRECL) 00000190
C                                                    00000210
C----- 2. SET CONTROL RECORD AND OUTPUT TO FILE NUNIT 00000220
C                                                    00000230
C     IWFILE = NUNIT 00000240
C     LCONTR=40 00000250
C     NCONTR=99 00000260
C     LBUFFR=1000 00000270
C     LRECOD=LENGTH 00000280
C     NRECOD=50000 00000290
C     IF(LRECOD.GT.LBUFFR) GO TO 3000 00000300
C     NUTOLD=0 00000310
C     NTOLD=0 00000320
C     DO 100 I=1,10 00000330
C     READ(BLANK,'(A4)') NODOLD(I,1) 00000340
C     NODOLD(I,2)=0 00000341
C     100 CONTINUE 00000342
C**** 2-1. SET TITLE 00000350
C     DO 2120 LP1 = 1,20 00000400
C       ICTR(LP1,IWFILE) = ITITLE(LP1) 00000410
C     2120 CONTINUE 00000420
C**** 2-2. SET AND INITIALIZE OTHER CONTROL VARIABLES 00000430
C     ICTR(21,IWFILE) = 2 00000440
C     ICTR(22,IWFILE) = 3 00000450
C     ICTR(23,IWFILE) = 2+NDIRCT 00000460
C     IF(CTR(23,IWFILE).LE.NRECOD) GO TO 2210 00000470
C       WRITE(6,7900) NUNIT,NRECOD 00000480
C       STOP 00000490
C     2210 CONTINUE 00000500
C     ICTR(24,IWFILE) = 0 00000510
C     ICTR(25,IWFILE) = 0 00000520
C     ICTR(26,IWFILE) = LRECOD 00000530
C     ICTR(27,IWFILE) = (LRECOD-4)/12 00000540
C     ICTR(28,IWFILE) = NDIRCT 00000550
C     ICTR(29,IWFILE) = NRECOD-NDIRCT-1 00000560

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        ICONTR(30,IWFILE) = 1          00000570
        ICONTR(31,IWFILE) = 0          00000580
        DO 2220 LP1 = 32,40          00000590
           ICONTR(LP1,IWFILE) = 0      00000600
2220 CONTINUE                                00000610
C**** 2-3. WRITE CONTROL SECTION            00000620
        WRITE(NUNIT,REC=1)  (ICONTR(LP1,IWFILE),LP1=1,LCONTR) 00000630
C   RESET NO. OF INITIALIZED RECORD          00000640
        IFLAG=0                      00000650
        DO 300 N=2,NRECOD            00000660
        READ(NUNIT,REC=N)
        IF(IFLAG,EQ,1)               GO TO 350      00000680
300 CONTINUE                                00000690
        N=NRECOD+1                  00000700
350 NRECOD=N-1                  00000710
        ICONTR(29,IWFILE)=NRECOD-NDIRCT-1    00000711
        WRITE(6,7000) NRECOD             00000720
        WRITE(6,7020)                  00000730
        WRITE(6,7030)  (ICONTR(LP1,IWFILE),LP1=1,18) 00000740
        WRITE(6,7040)  ICONTR(21,IWFILE)       00000750
        WRITE(6,7050)  ICONTR(22,IWFILE)       00000760
        WRITE(6,7060)  ICONTR(23,IWFILE)       00000770
        WRITE(6,7062)  ICONTR(24,IWFILE)       00000780
        WRITE(6,7064)  ICONTR(25,IWFILE)       00000790
        WRITE(6,7070)  ICONTR(26,IWFILE)       00000800
        WRITE(6,7080)  ICONTR(27,IWFILE)       00000810
        WRITE(6,7090)  ICONTR(28,IWFILE)       00000820
        WRITE(6,7100)  ICONTR(29,IWFILE)       00000830
        WRITE(6,7110)  ICONTR(30,IWFILE)       00000840
        WRITE(6,7120)  ICONTR(31,IWFILE)       00000850
        WRITE(NUNIT,REC=1)  (ICONTR(LP1,IWFILE),LP1=1,LCONTR) 00000870
C
C----- 3. MAKE FIRST LEVEL NODE DIRECTORY  00000880
C
        READ(BLANK,'(A4)')  IBUFFR(1)        00000890
        IBUFFR(2) = 0                   00000900
        IBUFFR(3) = 0                   00000910
        IBUFFR(4) = 0                   00000911
        IBUFFR(5) = 0                   00000930
        IBUFFR(6) = 0                   00000940
        IBUFFR(7) = 0                   00000950
        WRITE(NUNIT,REC=2)  (IBUFFR(LP1),LP1=1,4) 00000960
C
C----- 4. SET DATE                         00000970
C
        CALL DATE(NDATE)              00000980
        RETURN                         00000990
C
C----- 5. ERROR STOP                       00001000
C
        3000 WRITE(6,7800) LRECOD,LBUFFR      00001010
        STOP                           00001011
C
C----- 6. FORMAT                          00001012
C
        7000 FORMAT('*** MESSAGE FROM PINIT ***')
        *      '      NO. OF INITIALIZED RECORD IS ',15) 00001013
                                         00001014
                                         00001015
                                         00001020
                                         00001030
                                         00001040
                                         00001050
                                         00001060

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

7020 FORMAT(1H0,'***** C O N T R O L S E C T I O N *****'00001070
  *      /1H0,5X,'COL.')          00001080
7030 FORMAT(1H0,4X,'1-18 TITLE : '/1H0,6X,18A4)        00001090
7040 FORMAT(1H0,6X,'21 ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE 00001100
  *: ',I10)                      00001110
7050 FORMAT(1H0,6X,'22 HEAD ADDRESS FOR THE VACANT DIRECTORY AREA 00001120
  *: ',I10)                      00001130
7060 FORMAT(1H0,6X,'23 HEAD ADDRESS FOR THE VACANT DATA AREA 00001140
  *: ',I10)                      00001150
7062 FORMAT(1H0,6X,'24 WRITE FLAG 00001160
  *: ',I10)                      00001170
7064 FORMAT(1H0,6X,'25 READ FLAG (NOT USED) 00001180
  *: ',I10)                      00001190
7070 FDRHAT(1H0,6X,'26 LENGTH OF THE ONE PHYSICAL RECORD 00001200
  *: ',I10)                      00001210
7080 FORMAT(1H0,6X,'27 MAXIMUM NUMBER OF THE SAME LEVEL NODE 00001220
  *: ',I10)                      00001230
7090 FORMAT(1H0,6X,'28 SIZE OF THE DIRECTORY SECTION 00001240
  *: ',I10)                      00001250
7100 FORMAT(1H0,6X,'29 SIZE OF THE DATA SECTION 00001260
  *: ',I10)                      00001270
7110 FORMAT(1H0,6X,'30 REAL NUMBER OF THE DIRECTORY RECORDS 00001280
  *: ',I10)                      00001290
7120 FORMAT(1H0,6X,'31 REAL NUMBER OF THE DATA SET RECORDS 00001300
  *: ',I10)                      00001310
7800 FORHAT(1H ,10X,'***** PINIT ERROR *****'
  */15X,'RECORD LENGTH IS TOO LONG. LENGTH=',14,' MAX LENGTH=',I4) 00001320
7900 FORMAT(1H ,10X,'***** PINIT ERROR *****'
  */15X,'THE NUMBER OF DATA RECORD IS TOO LARGE      UNIT =',
  *13/15X,'TOTAL RECORD NO ',15)          00001331
                                         00001332
                                         00001340
                                         00001350
C                                         00001360
END

```

```
*****
** POPEN   **
*****
```

```

SUBROUTINE POPEN(NUNIT,JCONTR)          00000010
C                                         00000020
C----- 0. DECLARATION                  00000030
C                                         00000040
C     COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NSTAT 00000050
C                                         00000060
C     COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
*                     NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00000070
*                     NTHOLD,NODOLD(10,2),NRI1
*DIMENSION      JCONTR(1)                00000080
CHARACTER DDNAME*8,BLANK*4              00000090
DATA DDNAME//FT  F001//                 00000100
DATA BLANK//    ''/                      00000110
C                                         00000120
C----- 1. DECLARATION OF DIRECT ACCESS FILE 00000121
C                                         00000130
C                                         00000140
C                                         00000150
C     GET RECORD LENGTH OF THIS DATA POOL BEFORE OPEN 00000160
N1=NUNIT/10                            00000170

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

```

        WRITE(DDNAME(3:3),'(11)') N1          00000180
        N2=NUNIT-N1*10                      00000181
        WRITE(DDNAME(4:4),'(11)') N2          00000190
        CALL GETDCB(DDNAME,LRECL,LBLKS,RECFM,DSORG,IRCD)
        IF(IRCD.NE.0)                         GO TO 1000 00000200
        C OPEN DATA POOL                     00000210
           OPEN(NUNIT,ACCESS='DIRECT',RECL=LBLKS) 00000240
        C
        C----- 2. READ CONTROL SECTION AND SET DATE 00000251
        C
        LCONTR=40                           00000260
        IFILE=NUNIT                         00000270
        READ(NUNIT,REC=1) (ICONTR(1,IFILE),I=1,LCONTR) 00000280
        NCONTR=99                           00000281
        LBUFFR=1000                         00000282
        LRECOD=ICONTR(26,IFILE)              00000290
        NRECOD=1+ICONTR(28,IFILE)+ICONTR(29,IFILE) 00000310
        NUDOLD=0                            00000320
        NTOLD=0                             00000330
        DO 200 I=1,10                       00000340
           READ(BLANK,'(A4)') NUDOLD(I,1)    00000350
200  NUDOLD(I,2)=0                     00000360
        DO 300 L=1,LCONTR                 00000370
           JCONTR(L)=ICONTR(L,IFILE)       00000380
300  CONTINUE                         00000390
           CALL DATE(NDATE)               00000400
        C
        C----- 3. RETURN                  00000410
        C
        RETURN                             00000420
1000  WRITE(6,7000) DDNAME             00000430
        STOP                               00000440
        C
        C----- 4. FORMAT                  00000450
        C
        7000 FORMAT('0 **** POPEN ERROR   DDNAME=',A8,' IS NOT ALLOCATED') 00000460
        END                                00000470

```

```
*****
** PREAD  **
*****
```

```

        SUBROUTINE PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA) 00000010
        C
        C---- DECLARATION                   00000020
        C
        COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NSTAT 00000030
        DIMENSION ICH(1),NDATA(1)          00000040
        C
        C---- READ CONTROL SECTION OF SUB-DATA SET 00000050
        C
        IXSTAT = IX                        00000060
        READ(NUNIT,REC=IX) NAME1,NAME2,(ICH(LP1),LP1=1,20),NASBD,NOSBDS, 00000070
        *                                     NOARY,(NDATA(LP1),LP1=1,NOARY) 00000080
        C

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY)

TIME 15:00:36 PAGE 00

```

IX = IXSTAT          00000140
RETURN              00000150
END                 00000160

*****
** PREAD1  **
*****


      SUBROUTINE PREAD1(NUNIT,ICH,NDATA1,DATA1)          00000010
C-----          I00000020
C   1. FUNCTION          I00000030
C     (1) READ SUB DATA SET WHICH CONSISTS OF ONE ARRAY    I00000040
C                                         I00000050
C   2. INPUT          I00000060
C     (1) NUNIT      : FILE UNIT NO.                      (A)    I00000070
C     (2) IX         : READ-STARTING ADDRESS             (C/DPCONT/) I00000080
C     (3) ICH(20)    : THE TITLE OF DATA SET            (F NUNIT) I00000090
C     (4) NDATA1     : THE SIZE OF DATA ARRAY           (F NUNIT) I00000100
C     (5) DATA1      : THE DATA ARRAY                  (F NUNIT) I00000110
C                                         I00000120
C   3. OUTPUT          I00000130
C     (1) ICH(20)    I00000140
C     (2) NDATA1     I00000150
C     (3) DATA1      I00000160
C     (4) IX         : READ-STARTING POINT OF THE NEXT SUB    I00000170
C                         DATA SET                      I00000180
C-----          I00000190
C                                         00000200
C----- DECLARATION          00000210
C                                         00000220
C
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000230
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00000240
*          NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD, 00000250
*          NTHOLD,NODOLD(10,2),NA1 00000260
C                                         00000270
C
DIMENSION ICH(1),DATA1(1),NDATA(4)          00000280
C                                         00000290
C
IXOLD=IX          00000300
READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICH(LP1),LP1=1,20),NA1WK,NSDSWK,00000310
*          NOAWK,(NDATA(N),N=1,NOAWK), 00000320
*          (DATA1(LP1),LP1=1,NDATA(1)) 00000330
NDATA1=NDATA(1) 00000340
C
IF(NOAWK.EQ.1)      GO TO 1000 00000350
C
SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET 00000360
NWORD=25+NOAWK 00000370
DO 100 N=1,NOAWK 00000380
100 NWORD=NWORD+NDATA(N) 00000390
NRECD=NWORD/LRECOD 00000400
IF(NOD(NWORD,LRECOD).EQ.0)      GO TO 200 00000410
NRECD=NRECD+1 00000420
200 IX=IXOLD+NRECD 00000430
1000 RETURN 00000440
END 00000450

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```
*****
** PREAD2 **
*****
```

```

      SUBROUTINE PREAD2(NUNIT,ICM,NDATA1,DATA1,NDATA2,DATA2)      00000010
C                                                               00000020
C---- DECLARATION                                              00000030
C                                                               00000040
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT   00000050
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFRR(1000),NRECOD,NODE1,NODE2,
*                           NAOWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00000060
*                           NTHOLD,NODOLD(10,2),NA1                           00000070
*                                                               00000080
      DIMENSION ICM(1),DATA1(1),DATA2(1),NDATA(4)                  00000090
C                                                               00000100
C                                                               00000110
      IXOLD=IX
      READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICM(LP1),LP1=1,20),NA1WK,NSDSWK,00000130
*                           NOAWK,(NDATA(N),N=1,NOAWK),               00000140
*                           (DATA1(LP1),LP1=1,NDATA(1)),            00000150
*                           (DATA2(LP1),LP1=1,NDATA(2))              00000160
      NDATA1=NDATA(1)
      NDATA2=NDATA(2)
      IF(NOAWK.EQ.2)          GO TO 1000                         00000170
C SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET           00000180
      NWORD=25+NOAWK
      DO 100 N=1,NOAWK
 100 NWORD=NWORD+NDATA(N)
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 200                 00000190
      NRECD=NRECD+1
 200 IX=IXOLD+NRECD
 1000 RETURN
      END

```

```
*****
** PREAD3 **
*****
```

```

      SUBROUTINE PREAD3(NUNIT,ICM,NDATA1,DATA1,NDATA2,DATA2,NDATA3, 00000010
*                           DATA3)                                         00000020
C                                                               00000030
C---- DECLARATION                                              00000040
C                                                               00000050
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT   00000060
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFRR(1000),NRECOD,NODE1,NODE2,
*                           NAOWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00000070
*                           NTHOLD,NODOLD(10,2),NA1                           00000080
*                                                               00000090
      DIMENSION ICM(1),DATA1(1),DATA2(1),DATA3(1),NDATA(4)        00000100
C                                                               00000110
C                                                               00000120
      IXOLD=IX
      READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICM(LP1),LP1=1,20),NA1WK,NSDSWK,00000140
*                           NOAWK,(NDATA(N),N=1,NOAWK),               00000150
*                           (DATA1(LP1),LP1=1,NDATA(1)),            00000160

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

*          (DATA2(LP1),LP1=1,NDATA(2)),          00000170
*          (DATA3(LP1),LP1=1,NDATA(3))          00000180
*          NOATA1=NDATA(1)                      00000190
*          NOATA2=NDATA(2)                      00000200
*          NOATA3=NDATA(3)                      00000210
C          IF(NOAWK.EQ.3)                   GO TO 1000  00000220
C          SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET 00000230
C          NWORD=25+NOAWK                      00000240
DO 100 N=1,NOAWK                         00000250
100 NWORD=NWORD+NDATA(N)                  00000260
NRECD=NWORD/LRECOD                       00000270
IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 200  00000280
NRECD=NRECD+1                           00000290
200 IX=IXOLD+NRECD                      00000300
1000 RETURN                                00000310
END                                     00000320

*****
** PREAD4   **
*****
```

```

SUBROUTINE PREAD4(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,
*                      NDATA3,DATA3,NDATA4,DATA4)          00000010
*          NDATA3,DATA3,NDATA4,DATA4)          00000020
C
C---- DECLARATION
C
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT 00000030
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECD,NODE1,NODE2, 00000040
*          NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00000050
*          NTHOLD,N000LD(10,2),NA1 00000060
C
DIMENSION ICH(1),DATA1(1),DATA2(1),DATA3(1),DATA4(1),NDATA(4) 00000070
C
IXOLD=IX                                     00000080
READ(NUNIT,REC=IX) N00WK1,N00WK2,(ICH(LP1),LP1=1,20),NA1WK,NSDSWK,00000140
*          NOAWK,(NDATA(N),N=1,NOAWK),          00000090
*          (DATA1(LP1),LP1=1,NDATA(1)),          00000100
*          (DATA2(LP1),LP1=1,NDATA(2)),          00000110
*          (DATA3(LP1),LP1=1,NDATA(3)),          00000120
*          (DATA4(LP1),LP1=1,NDATA(4))           00000130
NDATA1=NDATA(1)                            00000140
NDATA2=NDATA(2)                            00000150
NDATA3=NDATA(3)                            00000160
NDATA4=NDATA(4)                            00000170
NWORD=25+NOAWK                           00000180
DO 100 N=1,NOAWK                         00000190
100 NWORD=NWORD+NDATA(N)                  00000200
NRECD=NWORD/LRECOD                       00000210
IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 200  00000220
NRECD=NRECD+1                           00000230
200 IX=IXOLD+NRECD                      00000240
1000 RETURN                                00000250
END                                     00000260
```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY)

TIME 15:00:36 PAGE 00

```
*****
** PRITE   **
*****
```

```

      SUBROUTINE PRITE(NUNIT,ICM)          00000100
C                                         00000200
C---- DECLARATION                      00000300
C                                         00000400
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NOSTAT 00000500
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFR(1000),NRECOD,NODE1,NODE2,
      *                               NADWN,NADAT,NOASET,NOATE(2),NINFOM(5),NUTOLD, 00000600
      *                               NTHOLD,NODOLD(10,2),NA1                         00000700
      COMMON /DPWCHK/ NRPEMT,NRWRTH,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3), 00000800
      *                               ISOBEF(12)                           00000900
C                                         00001000
C                                         00001100
      DIMENSION ICM(1)                     00001200
C                                         00001300
C---- CHECK SIZE                        00001400
C                                         00001500
      IXOLD=IX                            00001600
      IFILE = NUNIT                         00001700
C                                         00001800
      NWORD=25                            00001900
      NRECD = NWORD/LRECOD                00002000
      IF(NOD<NWORD,LRECOD).EQ.0)           GO TO 1020 00002100
      .NRECD = NRECD+1                     00002200
      1020 ICHKNO = NRPEMT-(NRWRTH+NRECD) 00002300
      IF(ICHKNO.GE.0)                      GO TO 2000 00002400
      CALL DCLEAR
      WRITE(6,7920)           (NODWTN(LP9,1),LP9=1,NTHWTN) 00002500
      GO TO 5200                           00002600
C                                         00002700
C---- WRITE DATA SET                   00002800
C                                         00002900
      2000 NSUBDS = NSUBDS+1               00003000
      NOA = 0                             00003100
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS,NOA 00003200
      IX = IX + NRECD                     00003300
C                                         00003400
C                                         00003500
C---- UPDATE CONTROL SECTION AND VARIABLES 00003600
C                                         00003700
      CALL WRTCHK(NUNIT,IXOLD)            00003800
C                                         00003900
C---- RETURN                           00004000
C                                         00004100
      S100 RETURN                         00004200
      S200 CONTINUE                         00004300
      ICTR(24,IFILE)=0                    00004400
      WRITE(NUNIT,REC=1) (ICTR(LP1,IFILE),LP1=1,LCONTR) 00004500
      STOP                                00004600
C                                         00004700
      7920 FORMAT(1H ,10X,'***** PRITE ERROR *****' 00004800
      */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN', 00004900
      */15X,10(A4,4X))                  00005000

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15 (THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

END

00005100

```
*****
** PRITE1 **
*****
```

```

      SUBROUTINE PRITE1(NUNIT,ICH,NDATA1,DATA1)          00000010
C                                         00000020
C---- DECLARATION                                00000030
C                                         00000040
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NOSTAT 00000050
      COMMON /DPWORK/ LBUFFR,LRECOO,IBUFR(1000),NRECOD,NODE1,NODE2, 00000060
      *             NAOWN,NADAT,NDASET,NDATE(2),NINFDM(5),NUTOLD, 00000070
      *             NTHOLD,NODOLD(10,2),NA1 00000080
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3), 00000090
      *             ISDBEF(12) 00000100
C                                         00000110
      DIMENSION ICH(1),DATA1(1)                      00000120
C                                         00000130
C---- CHECK SIZE .                                00000140
C                                         00000150
      IXOLD = IX 00000160
      IWFILE = NUNIT 00000170
      IF(NDATA1.GT.0) GO TO 1010 00000180
      WRITE(6,7910) (NODWTN(LP9,1),LP9=1,NTHWTN) 00000190
      CALL DCLEAR 00000200
      GO TO 5200 00000210
1010 NWORD = 26+NDATA1 00000220
      NRECD = NWORD/LRECOD 00000230
      IF(MOD(NWORD,LRECOD).EQ.0) GO TO 1020 00000240
      NRECD=NRECD+1 00000250
      1020 ICHKNO = NRPEMT-(NRWRTN+NRECD) 00000260
      IF(ICKHNO.GE.0) GO TO 2000 00000270
      CALL DCLEAR 00000280
      WRITE(6,7920) (NODWTN(LP9,1),LP9=1,NTHWTN) 00000290
      GO TO 5200 00000300
C                                         00000310
C---- WRITE DATA SET                            00000320
C                                         00000330
      2000 NSUBDS=NSUBDS+1 00000340
      NOA = 1 00000350
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICH(LP1),LP1=1,20),NA1,NSUBDS, 00000360
      *             NOA,NDATA1,(DATA1(LP1),LP1=1,NDATA1) 00000370
      IX = IX + NRECD 00000371
C                                         00000380
C---- UPDATE CONTROL SECTION AND VARIABLES 00000390
C                                         00000400
      CALL WRTCHK(NUNIT,IXOLD) 00000410
C                                         00000420
C---- RETURN                                     00000430
C                                         00000440
      5100 RETURN 00000450
      5200 CONTINUE 00000460
      1CTR(24,IWFILE)=0 00000470
      WRITE(NUNIT,REC=1) (ICCTR(LP1,IWFILE),LP1=1,LCTR) 00000480

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        STOP                               00000490
C
7910 FORMAT(1H ,10X,'***** PRITE1 ERROR *****'
           */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
           */15X,10(A4,4X))                00000500
           */15X,10(A4,4X))                00000510
           */15X,10(A4,4X))                00000520
           */15X,10(A4,4X))                00000530
7920 FORMAT(1H ,10X,'***** PRITE1 ERROR *****'
           */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
           */15X,10(A4,4X))                00000540
           */15X,10(A4,4X))                00000550
           */15X,10(A4,4X))                00000560
           END                               00000570

*****
** PRITE2 **
*****

      SUBROUTINE PRITE2(NUNIT,ICM,NDATA1,DATA1,NDATA2,DATA2)      00000010
C
C---- DECLARATION
C
COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT   00000020
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00000030
*          NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,          00000040
*          NTHOLD,NODOLD(10,2),NA1                           00000050
COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3), 00000060
*          ISDBEF(12)                                         00000070
*          ISDBEF(12)                                         00000080
*          ISDBEF(12)                                         00000090
*          ISDBEF(12)                                         00000100
*          ISDBEF(12)                                         00000110
C
DIMENSION ICM(1),DATA1(1),DATA2(1)                                00000120
C
C--- CHECK SIZE
C
IXOLO = IX
IWFIL = NUNIT
C
IF((NDATA1.GT.0).AND.(NDATA2.GT.0))      GO TO 1010             00000130
      WRITE(6,7910)      (NODWTN(LP9,1),LP9=1,NTHWTN)           00000140
      CALL DCLEAR
      GO TO 5200
1010 NWORD=27+NDATA1+NDATA2
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 1020             00000150
      NRECD=NRECD+1
1020 ICHKNO = NRPEMT-(NRWRTN+NRECD)          GO TO 2000           00000160
      IF(ICHKNO.GE.0)                  GO TO 2000           00000170
      CALL DCLEAR
      WRITE(6,7920)      (NODWTN(LP9,1),LP9=1,NTHWTN)           00000180
      GO TO 5200
C
C--- WRITE DATA SET
C
2000 NSUBDS=NSUBDS+1
      NOA=2
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS, 00000190
      *          NOA,NDATA1,NDATA2,(DATA1(LP1),LP1=1,NDATA1),       00000200
      *          (DATA2(LP1),LP1=1,NDATA2)                         00000210
      IX = IX + NRECD
C

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C---- UPDATE CONTROL SECTION AND VARIABLES          00000410
C                                                 00000420
C         CALL WRTCHK(NUNIT,IXOLD)                  00000430
C                                                 00000440
C---- RETURN                                     00000450
C                                                 00000460
C         5100 RETURN                                00000470
C         5200 CONTINUE                               00000480
C             I CONTR(24,IWFILE)=0                  00000490
C             WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
C             STOP                                    00000500
C                                                 00000510
C                                                 00000520
C         7910 FORMAT(1H ,10X,'***** PRITE2 ERROR *****'
C             */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
C             */15X,10(A4,4X))                      00000530
C             7920 FORMAT(1H ,10X,'***** PRITE2 ERRDR *****'
C             *15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
C             */15X,10(A4,4X))                      00000540
C             END                                     00000550
C                                                 00000560
C                                                 00000570
C                                                 00000580
C                                                 00000590

*****
** PRITE3 **
*****

SUBROUTINE PRITE3(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,NDATA3,
1                   DATA3)          00000010
C---- DECLARATION                                 00000020
C                                                 00000030
C                                                 00000040
C         COMMON /OPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C         COMMON /OPWORK/ LBUFFER,LRECOD,IBUFR(1000),NRECOD,NODE1,NODE2,
C                           * NADWN,NADAT,NDASET,NDATE(2),NINFO(5),NUTOLD,
C                           * NTHOLD,NODOLD(10,2),NA1
C         COMMON /OPWCHK/ NRPEMT,NWRWTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
C                           * ISOBEF(12)                         00000050
C                                                 00000060
C                                                 00000070
C                                                 00000080
C                                                 00000090
C                                                 00000100
C                                                 00000110
C         DIMENSION ICH(1),DATA1(1),DATA2(1),DATA3(1) 00000120
C                                                 00000130
C---- CHECK SIZE                                  00000140
C                                                 00000150
C         IXOLD=IX                                     00000160
C         IWFILE = NUNIT                                00000170
C                                                 00000180
C         IF((NDATA1.GT.0).AND.(NDATA2.GT.0).AND.(NDATA3.GT.0))
C             *                                         GO TO 1010 00000190
C             *                                         WRITE(6,7910)      (NODWTN(LP9,1),LP9=1,NTHWTN) 00000200
C             *                                         CALL DCLEAR
C             *                                         GO TO 5200                                00000210
C         1010 NWORD = 28+NDATA1+NDATA2+NDATA3           00000220
C         NRECD = NWORD/LRECOD                          00000230
C         IF(NODC(NWORD,LRECOD).EQ.0)                  GO TO 1020 00000240
C             *                                         NRECD = NRECD+1
C         1020 ICHKNO = NRPEMT-(NWRWTN+NRECD)          00000250
C         IF(ICKNO.GE.0)                                GO TO 2000 00000260
C             *                                         CALL DCLEAR
C             *                                         WRITE(6,7920)      (NODWTN(LP9,1),LP9=1,NTHWTN) 00000270
C                                                 00000280
C                                                 00000290
C                                                 00000300
C                                                 00000310

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT??

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

GO TO 5200
C
C---- WRITE DATA SET
C
2000 NSUBDS = NSUBDS+1
    NOA = 3
        WRITE(NUNIT,REC=1X) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS,
        *           NOA,NDATA1,NDATA2,NDATA3,(DATA1(LP1),LP1=1,NDATA1),
        *           (DATA2(LP1),LP1=1,NDATA2),(DATA3(LP1),LP1=1,NDATA3)
        IX = IX + NRECO
C
C---- UPDATE CONTROL SECTION AND VARIABLES
C
CALL WRTCHK(NUNIT,IXOLD)
C
C---- RETURN
C
5100 RETURN
5200 CONTINUE
    I CONTR(24,IWFILE)=0
    WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
    STOP
C
7910 FORMAT(1H ,10X,'***** PRITE3 ERROR *****'
        */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
        */15X,10(A4,4X))
7920 FORMAT(1H ,10X,'***** PRITE3 ERROR *****'
        */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
        */15X,10(A4,4X))
    END

*****
** PRITE4   **
*****


SUBROUTINE PRITE4(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,NDATA3,
*                           DATA3,NDATA4,DATA4)
C
C---- DECLARATION
C
COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
*                           NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,
*                           NTHOLD,NODOLD(10,2),NA1
COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
*                           ISDBEF(12)
C
DIMENSION ICM(1),DATA1(1),DATA2(1),DATA3(1),DATA4(1)
C
C---- CHECK SIZE
C
    IXOLD = IX
    IWFILE = NUNIT
C
IF((NDATA1.GT.0).AND.(NDATA2.GT.0).AND.(NDATA3.GT.0).AND.

```

FILE = J3679.DP0012.FORT77

DATE 84/03/15 (THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

        *(NDATA4.GT.0)
          WRITE(6,7910)      (NODWTH(LP9,1),LP9=1,NTHWTH)      GO TO 1010      00000210
          CALL DCLEAR
          GO TO 5200
1010 NWORD = 29+NDATA1+NDATA2+NDATA3+NDATA4      00000220
          NRECD = NWORD/LRECOD      00000230
          IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 1020      00000240
          NRECD = NRECD+1
1020 ICHKNO = NRPEHT-(NRWRDN+NRECD)      00000250
          IF(ICHKNO.GE.0)      GO TO 2000      00000260
          CALL DCLEAR
          WRITE(6,7920)
          GO TO 5200
C
C---- WRITE DATA SET
C
2000 NSUBDS = NSUBDS+1      00000270
  NOA = 4      00000280
  WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS,
  *      NOA,NDATA1,NDATA2,NDATA3,NDATA4,(DATA1(LP1),LP1=1,
  *      NDATA1),(DATA2(LP1),LP1=1,NDATA2),
  *      (DATA3(LP1),LP1=1,NDATA3),
  *      (DATA4(LP1),LP1=1,NDATA4)      00000290
  IX = IX + NRECD      00000300
C
C---- UPDATE CONTROL SECTION AND VARIABLES
C
  CALL WRTCHK(NUNIT,IXOLD)      00000310
C
C---- RETURN
C
5100 RETURN      00000320
5200 CONTINUE      00000330
  ICONTR(24,IWFILE)=0      00000340
  WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCTR)
  STOP      00000350
C
7910 FORMAT(1H ,10X,'***** PRITE4 ERROR *****'      00000360
  */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
  */15X,10(A4,4X))
7920 FORMAT(1H ,10X,'***** PRITE4 ERROR *****'      00000370
  */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
  */15X,10(A4,4X))
END      00000380

*****
** PSET **
*****
SUBROUTINE PSET(NUNIT,NODE,NTH,INFOH,NUPDAT,NRETUN)      00000390
C
C---- DECLARATION
C
  COMMON /DPCONT/ LCTR,NCTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00000400
  COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECD,NOOE1,NDREC,      00000410

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

*           NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,      00000700
*           NTHOLD,NODOLD(10,2),NA1                           00000800
COMMON /DPWCHK/ NRPEMT, NRWRTH, NIXOLD, NUTWTN, NTHWTN, NODWTN(10,3), 00000900
*           ISODEF(12)                                     00001000
C           DIMENSION NODE(1),NINFO(5),ISUBDR(12),NSDOLD(12)    00001100
C           EQUIVALENCE (NODE1,ISUBDR(1))                      00001200
C           00001300
C           00001400
C           00001500
C           00001600
C           00001700
C           IF(NTH.GT.0)                                     GO TO 1010 00001800
C           WRITE(6,7910)                                 GO TO 5100 00001900
C           00002000
1010 NSUBDS = 0                                         00002100
NRWRTH = 0                                         00002200
IWFIL = NUNIT                                     00002300
LRECOD = I CONTR(26,IWFIL)                         00002400
NIXOLD = I CONTR(23,IWFIL)                         00002500
NUTWTN = NUNIT                                     00002600
NTHWTN = NTH                                      00002700
DO 1020 LP1=1,NTH                                00002800
    NODWTN(LP1,1) = NODE(LP1)                         00002900
    NODWTN(LP1,2) = 2                               00003000
    NODWTN(LP1,3) = 2                               00003100
1020 CONTINUE                                     00003200
C**** SET FIRST LEVEL DIRECTORY ADDRESS AND CLEAR NODE COUNTER 00003300
IX = I CONTR(21,IWFIL)                            00003400
NCOUNT = 0                                         00003500
C           00003600
C           00003700
C           00003800
C           00003900
2000 NCOUNT = NCOUNT+1                            00004000
    READ(NUNIT,REC=IX)     (IBUFFR(LP1),LP1=1,LRECOD) 00004100
    IF(IBUFFR(4).GT.0)                                GO TO 2010 00004200
        IANDNM = 5                               00004300
        IWRK1 = IBUFFR(4)+1                         GO TO 2100 00004400
C**** SEARCH SUBDIRECTORY                         00004500
2010 IANDNM = 5                               00004600
    DO 2020 LP1 = 1,IBUFFR(A)                      00004700
        IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))       GO TO 3000 00004800
        IANDNM = IANDNM+12                         00004900
2020 CONTINUE                                     00005000
    IWRK1 = IBUFFR(4)+1                           00005100
C**** SUBDIRECTORY IS NOT FOUND                 00005200
2100 IF(IWRK1.LE.I CONTR(27,IWFIL))            GO TO 2110 00005300
    WRITE(6,7920)  (NODE(LP2),LP2=1,NTH)          00005400
    WRITE(6,7940)  NODE(NCOUNT)                   00005500
    GO TO 5100                                     00005600
2110 IF(NCOUNT.GE.NTH)                          GO TO 2600 00005700
C           00005800
C           00005900
C           00006000
C           00006100

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

2500 IBUFFR(4) = IWRK1          00006200
  NODE1 = NODE(NCOUNT)          00006300
  NDREC = 0                      00006400
  NADWN = ICNTR(22,IWFILE)      00006500
  NADAT = 0                      00006600
  NDASET = 0                     00006700
  DO 2510 LP1=1,5                00006800
2510 NINFOM(LP1)=0              00006900
  IWRK2=4+(IBUFFR(4)-1)*12       00007000
  DO 2520 LP1=1,12                00007100
    IWRK2=IWRK2+1                  00007200
2520  IBUFFR(IWRK2)=ISUBDR(LP1)  00007300
C   IX=IX-1                      00007400
  WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
  NODWTN(NCOUNT,2) = 0            00007600
                                GO TO 3200  00007700
C
C---- 2.6 ADD SUBDIRECTORY AND RETURN (NODE(NCOUNT) IS LAST NODE) 00007800
C
2600 IBUFFR(4) = IWRK1          00008100
  NODE1 = NODE(NCOUNT)          00008200
  NDREC = 0                      00008300
  NADWN = 0                      00008400
  NADAT = ICNTR(23,IWFILE)      00008500
  NDASET = 0                     00008600
C
  DO 2610 LP1=1,5                00008700
2610  NINFOM(LP1) = INFOM(LP1)  00008900
  IWRK2 = 4+(IBUFFR(4)-1)*12       00009000
  DO 2620 LP1=1,12                00009100
    IWRK2=IWRK2+1                  00009200
2620  IBUFFR(IWRK2) = ISUBDR(LP1)  00009300
C   IX=IX-1                      00009400
  NA1=IX                         00009500
  WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
  NODWTN(NCOUNT,2) = 0            00009600
  IX = NADAT                      00009700
  NRETUN = 0                      00009800
  NRPEMT = 1+ICNTR(28,IWFILE)+1CNTR(29,IWFILE)
*                           -(ICNTR(23,IWFILE)-1)  00010000
*                                         GO TO 5200  00010100
C
C---- 3. SUBDIRECTORY IS FOUND  00010200
C
3000 IF(NCOUNT.GE.NTH)          GO TO 4000  00010300
C*** NOT LAST NODE             00010400
  IWRK4=1ANDNM+2                  00010500
  IF(IBUFFR(IWRK4).EQ.0)          GO TO 3100  00010600
  IX = IBUFFR(IWRK4)               00010700
  GO TO 2000                      00010800
C*** THE DIRECTORY OF NODE(NCOUNT) IS NOT FOUND.  00010900
  3100 IBUFFR(IWRK4) = ICNTR(22,IWFILE)  00011000
C   IX=IX-1                      00011100
  WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
  NODWTN(NCOUNT,2) = 3            00011200
                                GO TO 5200  00011300
C

```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DP00L2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

C                               00011700
C---- MAKE DIRECTORY OF NODE(NCOUNT) 00011800
C                               00011900
C                               00012000
 3200 IWRK3=1+ICONTR(28,IWFILE)      GO TO 3210 00012100
     IF(ICONTR(22,IWFILE).LE.IWRK3)          00012200
C                               00012300
NTHWTN=NCOUNT                00012400
CALL DCLEAR                  00012500
WRITE(6,7930)  (NODE(LP9),LP9=1,NTH)    GO TO 5100 00012600
C                               00012700
C                               00012800
 3210 IBUFFR(1) = NODE(NCOUNT) 00012900
     IBUFFR(2) = 0
C                               00013000
     IBUFFR(3) = IX-1
     IBUFFR(3) = IX
     IBUFFR(4) = 0
C                               00013200
     DO 3220 LP1=5,LRECOD 00013400
 3220   IBUFFR(LP1)=0
     IX=ICONTR(22,IWFILE) 00013600
     WRITE(6,UNIT,REC=IX)  (IBUFFR(LP1),LP1=1,LRECOD) 00013700
     ICONTR(22,IWFILE) = ICONTR(22,IWFILE)+1 00013800
     ICONTR(30,IWFILE) = ICONTR(30,IWFILE)+1 00013900
     NODWTN(NCOUNT,3) = 0 00014000
C                               00014100
     IX=IX-1
                           GO TO 2000 00014200
C                               00014300
C---- 4. SUBDIRECTORY OF LAST NODE ALREADY EXIST. 00014400
C                               UPDATE SUBDIRECTORY OR IMMEDIATELY RETURN DEPENDING ON NUPDAT 00014500
C                               00014600
 4000 IWRK2 = IANDNM+3 00014700
     IF((NUPDAT.NE.0).OR.(IBUFFR(IWRK2).EQ.0))  GO TO 4010 00014800
     NRETUN = 1
                           GO TO 5200 00015000
C**** SAVE OLD SUBDIRECTORY TO ISDBEF(12) 00015100
 4010 IWRK3 = IANDNM - 1 00015200
     DO 4020 LP1=1,12 00015300
     IWRK3=IWRK3+1 00015400
     ISDBEF(LP1)=IBUFFR(IWRK3) 00015500
     NSDOLD(LP1)=IBUFFR(IWRK3) 00015600
 4020 CONTINUE 00015700
C**** UPDATE SUBDIRECTORY 00015800
  IWRK2=IANDNM+1 00015900
  IBUFFR(IWRK2)=0 00016000
  IWRK2=IWRK2+2 00016100
  IBUFFR(IWRK2) = ICONTR(23,IWFILE) 00016200
  IWRK2=IWRK2+1 00016300
  IBUFFR(IWRK2) = 0 00016400
  IWRK2=IWRK2+1 00016500
  IBUFFR(IWRK2) = NDATE(1) 00016600
  IWRK2=IWRK2+1 00016700
  IBUFFR(IWRK2) = NDATE(2) 00016800
  DO 4030 LP1=1,5 00016900
  IWRK2=IWRK2+1 00017000
  IBUFFR(IWRK2) = INFOM(LP1) 00017100

```

FILE = J3679.DP00L2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```

4030 CONTINUE                                00017200
C**** SET NEW SUBDIRECTORY TO ISUBDR(12)      00017300
  IWRKS = 1ANDNM-1                           00017400
  DO 4040  LP1=1,12                          00017500
    IWRKS=IWRKS+1                           00017600
    ISUBDR(LP1) = IBUFFR(IWRKS)              00017700
  4040 CONTINUE                                00017800
C                                         00017900
C     IX = IX-1                               00018000
  N1 = IX                                     00018100
  WRITE(NUNIT,REC=IX)   (IBUFFR(LP1),LP1=1,LRECO) 00018200
C**** UPDATE CONTROL VARIABLES             00018300
  NODWTN(NCOUNT,2) = 4                      00018400
  NRPEMT = 1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)-(ICONTR(23,IWFILE)) 00018500
  *          -1)                            00018600
  NRETUN = 0                                 00018700
C**** DELETE OLD DATA                      00018800
  CALL DATDLT(NUNIT,NSDOLD)                  00018900
C**** SET IX (WRITE STARTING POINT)       00019000
  IX = ICONTR(23,IWFILE)                    00019100
C                                         00019200
C---- RETURN                                00019300
C                                         00019400
  5200 RETURN                                00019500
  5100 ICONTR(24,IWFILE)=0                  00019600
  WRITE(NUNIT,REC=1)   (ICONTR(LP1,IWFILE),LP1=1,LCONTR) 00019700
  STOP.                                     00019800
C                                         00019900
C---- FORMAT                                00020000
C                                         00020100
  7910 FORMAT(1H ,10X,'***** PSET ERROR *****' 00020200
  */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0' ) 00020300
  7920 FORMAT(1H ,10X,'***** PSET ERROR *****' 00020400
  */15X,'THE NUMBER OF SUBDIRECTORY OF THE SAME LEVEL NODE IS TOO LAR00020500
  *GE'                                       00020600
  */15X,10(4X,A4))                         00020700
  7930 FORMAT(1H ,10X,'***** PSET ERROR *****' 00020800
  */15X,'DIRECTORY AREA CAN NOT BE OBTAINED' 00020900
  */15X,10(4X,A4))                         00021000
  7940 FORMAT(1H ,15X,'NODE NAME CAN NOT BE WRITTEN = ',A4) 00021100
  END                                         00021200

*****
** PSKIP   **
*****
```

```

SUBROUTINE PSKIP(NUNIT,NODS)                00000010
C SKIP OF SUB DATA SETS.                   00000020
C NODS IS NO. OF SKIPPED DATA SETS        00000030
  COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NSTAT 00000040
  COMMON /DPWORK/ LBUFFR,LRECO,IBUFFR(1000),NRECO,NODE1,NODE2, 00000050
  *           NADWN,NADAT,NDASET,NOATE(2),NINFOR(5),NUTOLD,
  *           NTHOLD,NODOLD(10,2),NA1               00000060
  *           DIMENSION ICH(20),NOATA(10)            00000070
  *           NREC=0                                00000080
  *           00000090
```

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15 (THURSDAY) TIME 15:00:36 PAGE 00

```

      IF(NODS.EQ.0)          GO TO 100      00000100
      IFILE = NUNIT           00000110
      IXLAST=1+ICONTR(28,IFILE)+ICONTR(29,IFILE) 00000120
10  IXOLD=IX               00000121
      READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICH(LP1),LP1=1,20),
      *                  HA1WK,NSDSWK,NOA,(NDATA(N),N=1,NOA) 00000130
      IF(NREC.EQ.0)          GO TO 20       00000140
      IF(NODWK1.NE.NODPRE) WRITE(6,7000) NODWK1 00000160
20  NODPRE=NODWK1          00000170
      HWORD=25+NOA           00000180
      DO 30 N=1,NOA          00000190
30  NWWORD=NWORD+NDATA(N) 00000200
      NRECD=NWORD/LRECD0    00000210
      IF(MOD(NWORD,LRECD).EQ.0) GO TO 40       00000220
      NRECD=NRECD+1          00000230
40  IX=IXOLD+NRECD        00000240
      IF(IX.GT.IXLAST)       GO TO 200      00000250
      NREC=NREC+1             00000260
      IF(NREC.LT.NODS)       GO TO 10       00000270
100 RETURN                00000280
200 WRITE(6,7010) IX      00000290
      RETURN                00000300
7000 FORMAT('0 **** WARNING IN PSKIP ****')
      *                   ' SKIPPED TO NEXT DATA SET (NODE NAME = ',A4,')' 00000320
7010 FORMAT('0 **** ERROR IN PSKIP ****')
      *                   ' DATA SET ADDRESS WAS OVERFLOWED (IX = ',I8,')' 00000340
      END :                  00000350

```

```

*****
** PWEND **
*****

```

```

      SUBROUTINE PWEND(NUNIT)          00000010
C
C----- 0. DECLARATION          00000020
C
      COMMON /DPCONT/ LCONT,NCNT,ICONTR(40,99),IX,NSUBDS,NDSTAT 00000030
      CHARACTER FNAME*40,BLANK*4          00000040
      DATA BLANK//      /
C
C----- 1. SET WRITE FLAG OFF AND UPDATE CONTROL SECTION 00000050
C
      DO 10 N=1,37,4          00000060
10  FNAME(N:N+3)=BLANK          00000082
      INQUIRE(NUNIT,NAME=FNAME)          00000083
      IFILE = NUNIT           00000090
      ICONTR(24,IFILE) = 0          00000100
      IXUSE=ICONTR(23,IFILE) - NDSTAT 00000110
      IXLAST=1+ICONTR(28,IFILE)+ICONTR(29,IFILE)-ICONTR(23,IFILE)+1 00000120
      WRITE(6,7000) NUNIT,FNAME,IXUSE,IXLAST 00000130
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IFILE),LP1=1,LCONT) 00000140
      RETURN                00000150
7000 FORMAT('0*** INFORMATION OF DATA POOL USAGE ***')
      *                   ' LOGICAL UNIT NO.      = ',16 /
      *                   ' DATA SET NAME       = ',A4/ 00000170
      *                   '                                     = ',A4/ 00000171

```

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15 (THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

*	'	NO. OF WRITTEN RECORDS = ',I6 /	000000172
*	'	REMAINS RECORDS = ',I6 )	00000180
END			00000190

\*\*\*\*\*  
\*\* PWSTAT \*\*  
\*\*\*\*\*

SUBROUTINE PWSTAT(NUNIT)	00000010
C	00000020
C----- 0. DECLARATION	00000030
C	00000040
C COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT	00000050
C	00000060
C----- 1. READ CONTROL SECTION AND CHECK WRITE FLAG	00000070
C	00000080
IWFILE = NUNIT	00000090
READ(NUNIT,REC=1,ERR=1110) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)	00000100
IF(ICONTR(24,IWFILE).NE.0)	GO TO 2010
	00000110
	GO TO 2110
1110 WRITE(6,7900) NUNIT	00000120
	GO TO 3100
C	00000130
C----- 2. REWRITE CONTROL SECTION OR JOB ABORT	00000140
C	00000150
2010 WRITE(6,7010) NUNIT	00000160
	GO TO 3100
2110 ICONTR(24,IWFILE) = 1	00000170
NDSTAT=ICONTR(23,IWFILE)	00000180
WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)	00000190
C	00000200
C----- 3. RETURN OR STOP	00000210
C	00000220
3000 RETURN	00000230
3100 STOP	00000240
C	00000250
C----- 4. FORMAT	00000260
C	00000270
7010 FORMAT(1H ,10X,'***** PWSTAT ABORT *****'	00000280
*/15X,'WRITE FLAG IS ALREADY ON UNIT = ',I2)	00000290
7900 FORMAT(1H ,10X,'***** PWSTAT ERROR *****'	00000300
*/15X,'CONTROL SECTION READ ERROR UNIT = ',I2)	00000310
END	00000320
	00000330
	00000340
	00000350

\*\*\*\*\*  
\*\* SETHSG \*\*  
\*\*\*\*\*

SUBROUTINE SETHSG(RET,ERRNO,N1,N2)	00000010
C	00000020
C SET ERROR SET INFORMATION IF ERROR OCCURED	00000030
C	00000040
COMMON /DPEHSG/ IFLAG,NERNO	00000050
INTEGER RET,ERRNO	00000060
IFLAG=1	00000070

FILE = J3679.DPOOL2.FORT77

DATE 84/03/15(THURSDAY)

TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

NERNO=ERRNO	00000080
RETURN	00000090
END	00000100

\*\*\*\*\*  
\*\* SUDDLT \*\*  
\*\*\*\*\*

SUBROUTINE SUDDLT(NUNIT,NSDOLD)	00000010
C	00000020
C---- DECLARATION	00000030
C	00000040
DIMENSION NSDOLD(1)	00000050
C	00000060
C---- DELETE DIRECTORY OF THE NODE WHICH HAS SUBDIRECTORY NSDOLD	00000070
C	00000080
CALL DIRDLT(NUNIT,NSDOLD)	00000090
C	00000100
C---- DELETE DATA OF THE NODE WHICH HAS SUBDIRECTORY NSDOLD	00000110
C	00000120
CALL DATDLT(NUNIT,NSDOLD)	00000130
RETURN	00000140
END	00000150

\*\*\*\*\*  
\*\* WRTCHK \*\*  
\*\*\*\*\*

SUBROUTINE WRTCHK(NUNIT,IXOLD)	00000100
C	00000200
C---- DECLARATION	00000300
C	00000400
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT	00000500
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,	00000600
* NADN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD,	00000700
* NTHOLD,NDODLD(10,2),NA1	00000800
COMMON /DPWCHK/ NRPEMT,NRWRTN,RIXOLD,NUTWTN,NTHWTH,NODWTH(10,3),	00000900
* ISOBEF(12)	00001000
C	00001100
C---- UPDATE CONTROL SECTION	00001200
C	00001300
IWFILE = NUNIT	00001400
ICONT(23,IWFILE) = IX	00001500
ICONT(31,IWFILE) = ICTR(31,IWFILE)+(IX-IXOLD)	00001600
C*** UPDATE NRWRTN AND SAVE WRITE STARTING POINT	00001700
NRWRTN = NRWRTN+(IX-IXOLD)	00001800
IXNEXT = IX	00001900
C	00002000
C---- UPDATE SUBDIRECTORY	00002100
C	00002200
READ(NUNIT,REC=NA1) (IBUFFR(LP1),LP1=1,LRECOD)	00002300
C*** SEARCH SUBDIRECTORY	00002400
IANDNM=S	00002500
DO 1030 LP1=1,IBUFFR(4)	00002600
IF(NODE1.EQ.IBUFFR(IANDNM)) GO TO 1100	00002700

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

FILE = J3679.DPOOL2.FORT77 DATE 84/03/15(THURSDAY) TIME 15:00:36 PAGE 00

```
IANDNM = IANDNM+12          00002800
1030 CONTINUE                00002900
      WRITE(6,7910)              00003000
                                GO TO 2200
***** UPDATE THE NUMBER OF SUB DATA SET
1100 IWRK = IANDNM + 1        00003100
      IBUFFR(IWRK) = NRWRTH     00003200
      IWRK = IWRK + 3           00003300
      IBUFFR(IWRK) = NSUBDS    00003400
      NDASET = NSUBDS          00003500
      WRITE(NUNIT,REC=NA1)      00003600
      (IBUFFR(LP1),LP1=1,LRECOD)
      IX = IXNEXT               00003700
C                               00003800
      2100 RETURN                00003900
      2200 STOP                  00004000
C                               00004100
      7910 FORMAT(1H ,10X,'***** WRTCHK ERROR *****'
      *!/15X,'SUB-DIRECTORY CAN NOT BE FOUND')
C                               00004200
C                               00004300
      END                         00004400
                                00004500
                                00004600
                                00004700
```

FILE = J3679.DPOOL2.ASM

DATE 84/03/14(WEDNESDAY) TIME 19:35:11 PAGE 0

## DIRECTORY LIST OF J3679.DPOOL2.ASM

***** MEMBER NAME	***** PAGE NO.	***** NO. OF CARDS
----------------------	-------------------	-----------------------

GETDCB	0007	150
--------	------	-----

```
*****
** GETDCB **
*****  

*****  

*          ***** 00000100  

*          * 00000200  

*          * 00000300  

*          * 00000400  

*          * 00000500  

*          * 00000600  

*          * 00000700  

*          * 00000800  

*          * 00000900  

*          * 00001000  

*          * IF 'XXXX' UNKNOWN 00001100  

*          * DSORG C*4 OUTPUT * 00001200  

*          *           IF 'XXXX' UNKNOWN * 00001300  

*          * RCD    I*4 OUTPUT * 00001400  

*          *           IF RCD=0 NORMAL END * 00001500  

*          *           IF RCD=8 DD MISSING * 00001600  

***** 00001700
```

FILE = J3679.DPOOL2.ASM

DATE 84/03/14(WEDNESDAY) TIME 19:35:11 PAGE 0

FILE = J3679.DPOOL2.ASH

DATE 84/03/14(WEDNESDAY)

TIME 19:35:11 PAGE 0

GETDCB	CSECT		00001800
	CNOP	0,4	00001900
	STH	14,12,12(13)	00002000
	LR	10,15	00002100
	USING	GETDCB,10	00002200
	ST	13,SAVE+4	00002300
	LR	12,13	00002400
	LA	13,SAVE	00002500
	ST	13,8(12)	00002600
*----- LOAD ARGUMENTS ADDR. 00002700			
	LM	2,7,0(1)	00002800
	SR	8,8	00002900
*----- ISSUE RDJFCB MACRO 00003000			
	MVC	DCB+40(8),0(2)	00003100
	RDJFCB	DCB	00003200
	LTR	15,15	00003300
	BNZ	ERRSET8	00003400
*----- ISSUE OBTAIN MACRO 00003500			
	MVC	DSNAME(44),JFCB	00003600
	MVC	VOLSER(6),JFCB+118	00003700
	OBTAIN	EXPARM	00003800
	LTR	15,15	00003900
	BNZ	ERRSET8	00004000
*----- GET LRECL 00004100			
	LH	9,JFCB+104	FROM JFCB 00004200
	ST	9,0(3)	00004300
	LTR	9,9	00004400
	BNZ	JUMP1	00004500
	LH	9,WORK+44	FROM DSCB 00004600
	ST	9,0(3)	00004700
*----- GET BLKSIZE 00004800			
JUMP1	LH	9,JFCB+102	FROM JFCB 00004900
	ST	9,0(4)	00005000
	LTR	9,9	00005100
	BNZ	JUMP2	00005200
	LH	9,WORK+42	FROM DSCB 00005300
	ST	9,0(4)	00005400
*----- GET RECFM 00005500			
JUMP2	MVC	BYTE1(1),JFCB+100	FROM JFCB 00005600
	CLI	BYTE1,X'00'	00005700
	BNE	JUMP3	00005800
	MVC	BYTE1(1),WORK+40	FROM DSCB 00005900
JUMP3	LA	0,17	00006000
	LA	1,RECFM	00006100
LOOP1	CLC	BYTE1(1),0(1)	00006200
	BE	JUMP4	00006300
	A	1,F5	00006400
	BCT	0,LOOP1	00006500
JUMP4	MVC	0(4,5),1(1)	00006600
*----- GET DSORG 00006700			
	MVC	BYTE1(1),JFCB+98	FROM JFCB 00006800
	CLI	BYTE1,X'00'	00006900
	BNE	JUMPS	00007000
	MVC	BYTE1(1),WORK+38	FROM DSCB 00007100
JUMPS	LA	0,4	00007200

FILE = J3679.DPOOL2.ASH

DATE 84/03/14(WEDNESDAY)

TIME 19:35:11 PAGE 0

FILE = J3679.DPOOL2.ASM DATE 84/03/14(WEDNESDAY) TIME 19:35:11 PAGE 0

	LA	1,DSORG	00007300
LOOP2	CLC	BYTE1(1),0(1)	00007400
	BE	JUMP6	00007500
	A	1,F5	00007600
	BCT	0,LOOP2	00007700
JUMP6	CLI	0(1),X'02'	00007800
	BNE	JUMP7	00007900
	CLC	JFCB+44(8),BLANK	00008000
	BE	JUMP7	00008100
	LA	1,DSORG+5	00008200
JUMP7	MVC	0(4,6),1(1)	00008300
	B	RETURN	00008400
*****			
*			* 00008500
*			* 00008600
*		RETURN & ERRSET	* 00008700
*			* 00008800
*****			
ERRSET8	A	8,F8	00008900
RETURN	EQU	*	00009000
	ST	8,0(7)	00009100
	L	13,SAVE+4	00009200
	LM	14,12,12(13)	00009300
	SR	15,15	00009400
	BR	14	00009500
			00009600
*****			
*			* 00009700
*			* 00009800
*		DCB EXIT	* 00009900
*			* 00010000
*****			
	OS	OF	00010100
JFCBEXIT	DC	X'07',AL3(JFCB)	00010200
	DC	X'80',AL3(0)	00010300
			00010400
*****			
*			* 00010500
*			* 00010600
*		DC & DS	* 00010700
*			* 00010800
*****			
SAVE	DS	18F	00010900
F5	DC	F'5'	00011000
F8	DC	F'8'	00011100
DCB	DCB	DSORG=PS,HACRF=R,EXLST=JFCBEXIT,DDNAME=DUMMY	00011200
JFCB	DS	OD	00011300
	DC	CL176' ,	00011400
EXPARM	DS	OF	00011500
	DC	XL4'C100000'	00011600
	DC	A(DSNAME)	00011700
	DC	A(VOLSER)	00011800
	DC	A(WORK)	00011900
VOLSER	DC	CL6' ,	00012000
DSNAME	DC	CL44' ,	00012100
BLANK	DC	CL8' ,	00012200
BYTE1	DC	CL1' ,	00012300
WORK	DS	OD	00012400
	DC	CL148' ,	00012500
RECFM	DC	X'40',C'V'	00012600
			00012700

FILE = J3679.DPOOL2.ASM DATE 84/03/14(WEDNESDAY) TIME 19:35:11 PAGE 0

FILE = J3679.DPOOL2.ASM

DATE 84/03/14(WEDNESDAY) TIME 19:35:11 PAGE 0

DC	X'50',C'VB	'	00012800	
DC	X'44',C'VA	'	00012900	
DC	X'54',C'VBA	'	00013000	
DC	X'52',C'VBH	'	00013100	
DC	X'48',C'VS	'	00013200	
DC	X'58',C'VBS	'	00013300	
DC	X'80',C'F	'	00013400	
DC	X'90',C'FB	'	00013500	
DC	X'88',C'FS	'	00013600	
DC	X'98',C'FBS	'	00013700	
DC	X'84',C'FA	'	00013800	
DC	X'94',C'FBA	'	00013900	
DC	X'20',C'D	'	00014000	
DC	X'30',C'DB	'	00014100	
DC	X'CO',C'U	'	00014200	
DC	X'C4',C'UA	'	00014300	
DC	X'00',C'XXXX'		00014400	
DSORG	DC	X'80',C'IS	'	00014500
	DC	X'40',C'PS	'	00014600
	DC	X'20',C'DA	'	00014700
	DC	X'02',C'PO	'	00014800
	DC	X'00',C'XXXX'		00014900
	END		00015000	

## Appendix C Program List of POOL

## DIRECTORY LIST OF J3679.POOL2.FORT77

*****	*****	*****
MEMBER NAME	PAGE NO.	NO. OF CARDS
*****	*****	*****
(NO.=001) CATL	0001	11
(NO.=002) CONDENSE	0001	59
(NO.=003) COPY	0002	97
(NO.=004) DELETE	0004	22
(NO.=005) FLAG	0004	22
(NO.=006) HELP	0005	27
(NO.=007) INIT	0005	26
(NO.=008) LIST	0006	62
(NO.=009) MEND	0007	348
(NO.=010) HTCOPY	0014	44
(NO.=011) HTSAVE	0014	54
(NO.=012) RENAME	0016	91
(NO.=013) TREE	0017	41
-----		
904 CARDS		

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0001

```
*****
** CATAL   **
*****
```

```
C*****C00000100
C*          *C00000200
C*          CATAL      *C00000300
C*          *C00000400
C*****C00000500
PROGRAM CATAL          00000600
DIMENSION LCONTR(40)      00000700
CALL POPEN(91,LCONTR)      00000800
CALL CATLST(91)           00000900
STOP                      00001000
END                       00001100
```

```
*****
** CONDENSE **
*****
```

```
C*****C00000100
C*          *C00000200
C*          CONDENSE    *C00000300
C*          *C00000400
C*****C00000500
PROGRAM COND             00000600
DIMENSION JCONTR(40),NODE(10),IDATE(2) 00000700
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000800
C          00000900
COMMON /DPWORK/ LOUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00001000
*          NADWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD, 00001100
*          NTHOLD,NODOLD(10,2),NA1 00001200
COMMON /DPCUNT/ NOREC,NONOD 00001300
CHARACTER*24 DSN,BLK24 00001400
CHARACTER*4 BLANK 00001500
DATA BLK24,BLANK/' 00001600
ITP1=1 00001700
NUNIT=91 00001800
DSN=BLK24 00001900
CALL POPEN(NUNIT,JCONTR) 00002000
CALL PWSTAT(NUNIT) 00002100
NTH=0 00002200
NOREC=0 00002300
NONOD=0 00002400
CALL DATE(IDATE) 00002500
REWIND 1 00002600
WRITE(1) IDATE,(ICONTROL(I,NUNIT),I=1,20),(ICONTROL(I,NUNIT),I=26,31) 00002700
IX=ICONTROL(21,NUNIT) 00002800
KEY=1 00002900
CALL PHNLST(NUNIT,NODE,NTH,KEY,LLL) 00003000
IF(LLL.EQ.1) GO TO 1000 00003100
C. BACK-UP DATA WAS STORED ON ITP1 00003200
INQUIRE(ITP1,NAME=DSN) 00003300
WRITE(6,6000) DSN 00003400
```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0001

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0002

```

C RESET CONTROL SECTION          00003500
ICONTR(22,NUNIT)=3              00003600
ICONTR(23,NUNIT)=ICONTR(28,NUNIT)+2 00003700
ICONTR(24,NUNIT)=1              00003800
ICONTR(30,NUNIT)=1              00003900
ICONTR(31,NUNIT)=0              00004000
NUTOLD=0                        00004100
NTIHOLD=0                        00004200
NDSTAT=ICONTR(23,NUNIT)          00004300
WRITE(NUNIT,REC=1) (ICONTR(LP1,NUNIT),LP1=1,LCONTR) 00004400
READ(BLANK,'(A4)') IBUFFR(1)    00004500
READ(BLANK,'(A4)') IBUFFR(2)    00004600
IBUFFR(3)=0                      00004700
IBUFFR(4)=0                      00004800
WRITE(NUNIT,REC=2) (IBUFFR(LP1),LP1=1,4) 00004900
C LOAD BACK-UP DATA            00005000
REWIND 1                         00005100
READ()                           00005200
CALL PRECVR(NUNIT,2,LLL)         00005300
CALL PWEND(NUNIT)                00005400
C                               00005500
1000 STOP                         00005600
C                               00005700
6000 FORMAT(' BACK-UP DATA SET ',A24,'WAS CREATED') 00005800
END                             00005900

```

\*\*\*\*\*  
\*\* COPY \*\*  
\*\*\*\*\*

```

*****00000100
C* *00000200
C*      COPY *00000300
C*                                *00000400
*****00000500
PROGRAM COPY 00000600
DIMENSION JCONTR(40),NDIRC(12) 00000700
COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT 00000800
C 00000900
COMMON /DPWORK/ LBUFFER,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00001000
* NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD, 00001100
* NTHOLD,NODOLD(10,2),NA1 00001200
COMMON /DPCUNT/ NOREC,NONOD 00001300
CHARACTER*4 NODE(10)           00001400
CHARACTER*24 DSN2,BLK24        00001500
DATA BLK24/'                  00001600
C 00001700
DSN2=BLK24                     00001800
NUNIT=91                        00001900
IUNIT=92                        00002000
CALL POPEN(NUNIT,JCONTR)        00002100
CALL POPEN(IUNIT,JCONTR)        00002200
INQUIRE(IUNIT,NAME=DSN2)        00002300
50 WRITE(6,6000)                00002400
100 CALL NODINP(NTH,NODE)       00002500

```

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0002

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0003

```

IF(NTH.EQ.0) GO TO 1000 00002600
NDREC=0 00002700
NONOD=0 00002800
NOUP =0 00002900
REWIND 1 00003000
REWIND 2 00003100
IF(NODE(1).NE.'*ALL') GO TO 110 00003200
WRITE(6,6010) 00003300
IX=ICONTR(21,IUNIT) 00003400
NTH=0 00003500
GO TO 150 00003600
110 NTH1=NTH-1 00003700
IF(NTH1.EQ.0) GO TO 130 00003800
C PROCESS FOR UPPER NODE 00003900
DO 120 N=1,NTH1 00004000
CALL PFIND(NUNIT,NODE,N,NDIRC,LLL) 00004100
IF(LLL.EQ.0) GO TO 120 00004200
WRITE(6,6020) 00004300
GO TO 100 00004400
120 CALL PSAVE(NUNIT,NODE,N,NDIRC,LLL) 00004500
NOUP =NTH1 00004600
130 CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL) 00004700
IF(LLL.EQ.0) GO TO 140 00004800
WRITE(6,6020) 00004900
GO TO 100 00005000
140 CALL PSAVE(NUNIT,NODE,NTH,NDIRC,LLL) 00005100
IX=NDIRC(3) 00005200
IF(IX.EQ.0) GO TO 200 00005300
00005400
C
C PROCESS FOR LOWER NODES 00005500
150 KEY=1 00005600
CALL PNLIST(NUNIT,NODE,NTH,KEY,LLL) 00005700
IF(LLL.EQ.1) GO TO 1000 00005800
00005900
C
C CHECK NODE NAMES IF EXIST ON IUNIT DATA POOL 00006000
200 IF(ICONTR(30,IUNIT).EQ.1 .AND. ICONTR(31,IUNIT).EQ.0) 00006100
* GO TO 300 00006200
00006300
REWIND 2 00006400
NERR=0 00006500
DO 250 N=1,NONOD 00006600
READ(2) NO,(NODE(I),I=1,NO) 00006700
IF(N.LE.NOUP) GO TO 250 00006800
CALL PFIND(IUNIT,NODE,NO,NDIRC,LLL) 00006900
IF(LLL.NE.0) GO TO 250 00007000
WRITE(6,6030) DSN2,(NODE(I),I=1,NO) 00007100
NERR=NERR+1 00007200
250 CONTINUE 00007300
IF(NERR.EQ.0) GO TO 300 00007400
WRITE(6,6040) 00007500
GO TO 100 00007600
C
C START COPY 00007700
300 REWIND 1 00007800
CALL PWSTAT(IUNIT) 00007900
CALL PRECVR(IUNIT,2,LLL) 00008000

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0003

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0004

```

        CALL PWEND(IUNIT)                      00008100
C*                                         00008200
        WRITE(6,6050)                         00008300
        IF(NTH.EQ.0)                           GO TO 1000 00008400
        WRITE(6,6060)                         00008500
        GO TO 100                            00008600
1000 STOP                                00008700
6000 FORMAT(' ENTER NODE NAME. IF *ALL IS ENTERD, ALL DATA IS COPIED') 00008800
6010 FORMAT(' ALL DATA IS COPIED')          00008900
6020 FORMAT(' INPUT NODE NAME IS NOT FOUND.'// PLEASE RE-ENTER NODE NAME00009000
*E')
6030 FORMAT(' EXIST SAME NODE NAME IN A DATA POOL ',A24) 00009200
*   // NODE NAME = ',A4,9(.,,A4))           00009300
6040 FORMAT(' COPY IS NOT EXECUTED. PLEASE RE-ENTER')      00009400
6050 FORMAT(' DATA COPY WAS FINISHED SUCCESSFULLY')       00009500
6060 FORMAT(' ENTER NEXT NODE NAME')            00009600
END                                     00009700

*****
** DELETE **
*****

C*****C00000100
C*                                         *C00000200
C*                                         DELETE *C00000300
C*                                         *C00000400
C*****C00000500
PROGRAM DELETE                           00000600
DIMENSION NODE(10),LCONTR(40)           00000700
CALL POPEN(91,LCONTR)                   00000800
1 CONTINUE                               00000900
WRITE(6,300)                            00001000
CALL NODINP(NTH,NODE)
IF(NTH.EQ.0) GO TO 500                 00001100
CALL PDELT(91,NODE,NTH,NRETRN)         00001200
IF(NRETRN.EQ.0) WRITE(6,200) (NODE(I),I=1,NTH) 00001300
IF(NRETRN.NE.0) WRITE(6,250) (NODE(I),I=1,NTH) 00001400
GO TO 1                                 00001500
500 CONTINUE                            00001600
STOP                                     00001700
200 FORMAT(' NORMAL RETURN *** NODE NAME = ',A4,9(.,,A4)) 00001800
250 FORMAT(' ABNORMAL RETURN *** NODE NAME = ',A4,9(.,,A4)) 00001900
300 FORMAT(' ENTER NODE NAME.')          00002000
END                                     00002100
00002200

*****
** FLAG **
*****

C*****C00000100
C*                                         *C00000200
C*                                         FLAG *C00000300
C*                                         *C00000400
C*****C00000500
PROGRAM FLAG                           00000600

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0004

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0005

```

DIMENSION JCONTR(40)                                00000700
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NOSTAT   00000800
COMMON /DPWORK/ LBUFFER,LRECOD,IBUFR(1000),NRECOD,NODE1,NODE2, 00000900
*          NADWN,NADAT,NASET,NDATE(2),NINFOR(S),NUTOLD,        00001000
*          NTHOLD,NOOOLD(10,2),NA1                         00001100
C                                         00001200
      NUNIT=91                                         00001300
      CALL POPEN(NUNIT,JCONTR)                          00001400
      WRITE(6,100) ICTR(24,NUNIT)                      00001500
      ICTR(24,NUNIT)=0                               00001600
      WRITE(6,200)                                     00001700
      WRITE(NUNIT,REC=1) (ICTR(I,NUNIT),I=1,LRECOD)    00001800
      STOP                                         00001900
100 FORMAT(' CURRENT STATUS OF WRITE FLAG = ',13) 00002000
200 FORMAT(' NOW CHANGE WRITE FLAG TO 0')        00002100
      END                                         00002200

```

\*\*\*\*\*  
\*\* HELP \*\*  
\*\*\*\*\*

```

*****  

C                                         * 00000010
C                                         * 00000020
C             HELP                           * 00000030
C                                         * 00000040
*****  

      PROGRAM HELP                         00000050
C                                         00000060
      WRITE(6,6000)                         00000070
C                                         00000080
      STOP                                     00000090
C                                         00000100
      6000 FORMAT(' COMMAND           CONTENTS'// 00000110
      *     ' CATL   PRINT OF CONTROL AND DIRECTORY SECTION'// 00000120
      *     ' CONDENSE CONDENSE OF A DATA POOL'// 00000130
      *     ' COPY    COPY OF A NODE DATA'// 00000140
      *     ' DELETE  DELETE OF A NODE DATA'// 00000150
      *     ' FLAG    CHANGE OF A WRITE FLAG'// 00000160
      *     ' INIT   INITIALIZATION OF A DATA POOL'// 00000170
      *     ' LIST   LISTING OF A NODE DATA (SUB-DIRECTORY AND FORM 00000180
      *OF DATA ARRAYS)'// 00000190
      *     ' MEND   MENDING OF A CONTROL, DIRECTORY AND DATA COMMEN00000210
      *T'/'  ' MTCOPY  LOAD OF A BACK-UP TAPE TO A DATA POOL'// 00000220
      *     ' HTSAVE  MAKING OF A BACK-UP TAPE'// 00000230
      *     ' RENAME  RENAME OF A NODE'// 00000240
      *     ' TREE   PRINT OF ALL NODE NAMES IN A DATA POOL BY A TRE00000250
      *E STRUCTURE')                         00000260
      END                                         00000270

```

\*\*\*\*\*  
\*\* INIT \*\*  
\*\*\*\*\*

```

*****C00000100
C*                                         *C00000200

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0005

FILE = J3679-POOL2.EOR177 DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0006

```

C*           INIT
C*
C***** ****
      PROGRAM     INIT
      CHARACTER*80 IREC,TITLE
      CHARACTER*8 NDATE,DLANK
      DATA BLANK/'      /
10   WRITE(6,300)
      READ(5,100) IREC
      NDIRCT=ICCONV(IREC)
      IF(NDIRCT.EQ.-1)          GO TO 10
      WRITE(6,400)
      READ(5,200) (TITLE(I:I),I=1,64)
      CALL DATE(NDATE)
      TITLE(65:72)=BLANK
      TITLE(73:80)=NDATE
      LREC=900
      CALL PINIT(91,NDIRCT,LREC,TITLE)
      STOP
100  FORMAT(A80)
200  FORMAT(64A1)
300  FORMAT(' ENTER DIRECTORY SIZE ')
400  FORMAT(' ENTER TITLE (64 CHARACTERS)')
      END

*****
** LIST   **
*****
C***** ****
      PROGRAM     LIST
      DIMENSION JCTR(40),NDIRC(12),NODE(10),ICH(20),
      *          NDATA(4)
      COMMON /DPCONT/ LCTR,NCTR,ICTR(40,99),IX,NSUBDS,NDSTAT
C
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
      *          NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOL0,
      *          NTHOLD,NODOLD(10,2),NA1
      NUNIT=91
      CALL POPEN(NUNIT,JCTR)
10   WRITE(6,6000)
20   CALL NODEINP(NTH,NODE)
      IF(NTH.EQ.0)          GO TO 1000
      CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
      IF(LLL.EQ.0)          GO TO 100
      WRITE(6,6010)
      GO TO 20
C  PRINT DIRECTORY
100  WRITE(6,6020) (NODE(N),N=1,NTH)
      CALL PRSUBD(NDIRC)
C  PRINT DATA

```

FILE = J3679.POOL2.FORT77 DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0006

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0007

```

      NDASET=NDIRC(5)
      IF(NDASET .EQ. 0)           GO TO 200          00002700
      DO 150 N=1,NDASET          00002800
      CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA)
      WRITE(6,6040) N,(ICH(I),I=1,18),NOARY          00002900
      IF(NOARY.EQ.0) THEN
         WRITE(6,6045)
      ELSE
         WRITE(6,6050) (I,NDATA(I),I=1,NOARY)        00003000
      END IF
C   SET ADDRESS FOR A NEXT DATA SET          00003100
      NWORD=25+NOARY          00003200
      DO 120 I=1,NOARY          00003300
      120 NWORD=NWORD+NDATA(I)          00003400
      NRECD=NWORD/LRECOD          00003500
      IF(MOD(NWORD,LRECOD).EQ.0)    GO TO 130          00003600
      NRECD=NRECD +1          00003700
      130 IX=IX + NRECD          00003800
      150 CONTINUE          00003900
      200 CONTINUE          00004000
      GO TO 10          00004100
C
      1000 STOP          00004200
C
      6000 FORMAT(' ENTER NODE NAME ')          00004300
      6010 FORMAT(' NOT FOUND THIS NODE NAME, PLEASE RE-ENTER NODE NAME') 00004400
      6020 FORMAT(' RECORD INFORMATIONS FOR NODE NAME ',A4,9(',',A4)) 00004500
      6030 FORMAT(' ADDRESS OF A LOWER NODE DIRECTORY ',I6
      *      '/ ADDRESS OF A DATA SET      ',I6          00004600
      *      '/ LENGTH OF A DATA SET ',I4,' NO. OF SUB-DATA SETS ',I4 00004700
      *      '/ DATA OF CREATION      ',2A4)          00004800
      6040 FORMAT(' ** INFORMATIONS FOR SUB-DATA SET ',I4,' **/4X,18A4 00004900
      *      '/ NO. OF DATA ARRAYS ',I2)          00005000
      6045 FORMAT('      NOTHING OF DATA ARRAY')          00005100
      6050 FORMAT((4X,2('LENGTH OF DATA ',I1,'+',I6,3X)))          00005200
      END          00005300

*****
** MEND **
*****
```

```

C*****00000010
C*          *00000020
C*          MEND          *00000030
C*          *00000040
C*****00000050
      PROGRAM MEND          00000060
      DIMENSION JCONTR(40)          00000070
      DIMENSION NODE(10),ICH(20),NDIRC(12),NDATA(4),NDAIX(200) 00000080
      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT 00000090
      0000100
C
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
      *          NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD, 00000110
      *          NTHOLD,NODOLD(10,2),NA1          00000120
      *          00000130
      CHARACTER REC*80,CDAT*4,SLASH*4          00000140

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0007

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0008

```

DIMENSION RBUFFR(1)                                00000150
EQUIVALENCE (IBUFFR(1),RBUFFR(1))                00000160
DATA SLASH/////////                                00000170
C                                                 00000180
NUNIT=91                                         00000190
CALL POPEN(NUNIT,JCONTR)                         00000200
IDX=ICONTR(22,NUNIT)-1                           00000210
WRITE(6,6000) (ICONTR(J,NUNIT),J=1,18),ICONTR (21,NUNIT),IDX,
+                                              ICONTR(30,NUNIT),ICONTR(24,NUNIT) 00000220
IF(ICONTR(24,NUNIT).EQ.0)                         GO TO 5   00000230
WRITE(6,6010)                                     00000240
READ( 5,5000,END=2000) REC                      00000250
IF(REC(1:2).EQ.'OK')                            GO TO 5   00000260
GO TO 2000                                       00000270
5  ICONTR(24,NUNIT)=1                           00000280
WRITE(NUNIT,REC=1) (ICONTR(J,NUNIT),J=1,LCONTR)  00000290
10 WRITE(6,6100)                                     00000300
20 READ(5,5000,END=1000) REC                     00000310
IF(REC(1:4).EQ.'CONT')                           GO TO 100 00000320
IF(REC(1:5).EQ.'DIREC')                          GO TO 200 00000330
IF(REC(1:3).EQ.'COM')                            GO TO 500 00000340
IF(REC(1:3).EQ.'END')                            GO TO 1000 00000350
WRITE(6,6110)                                     00000360
GO TO 20                                         00000370
C MEND CONTROL SECTION                         00000380
100 N=0                                         00000390
WRITE(6,6120)                                     00000400
WRITE(6,6125) (ICONTR(LP1,NUNIT),LP1=1,18)      00000410
WRITE(6,6130) ICONTR(21,NUNIT)                  00000420
WRITE(6,6135) ICONTR(22,NUNIT)                  00000430
WRITE(6,6140) ICONTR(23,NUNIT)                  00000440
WRITE(6,6145) ICONTR(24,NUNIT)                  00000450
WRITE(6,6150) ICONTR(25,NUNIT)                  00000460
WRITE(6,6155) ICONTR(26,NUNIT)                  00000470
WRITE(6,6160) ICONTR(27,NUNIT)                  00000480
WRITE(6,6165) ICONTR(28,NUNIT)                  00000490
WRITE(6,6170) ICONTR(29,NUNIT)                  00000500
WRITE(6,6175) ICONTR(30,NUNIT)                  00000510
WRITE(6,6180) ICONTR(31,NUNIT)                  00000520
110 WRITE(6,6200)                                 00000530
120 READ(5,5000) REC                           00000540
KC=ICCONV(REC)
IF(KC.EQ.0)                                     GO TO 150 00000550
IF(KC.GE.1 .AND. KC.LE.31)                      GO TO 130 00000560
WRITE(6,6210)                                     00000570
GO TO 120                                         00000580
130 IF(KC.EQ.1) THEN
    WRITE(6,6215)
    READ(5,5100) (ICH(I),I=1,20)               00000590
    DO 140 I=1,20
140  ICONTR(I,NUNIT)=ICH(I)                      00000600
ELSE
    WRITE(6,6220)
    READ(5,5000) REC                           00000610
    IREC=ICCONV(REC)                         00000620

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0008

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0009

```

        IF(IREC.LT.0) THEN          00000700
          WRITE(6,6230)
          GO TO 120
        END IF
        ICONT(R(KC,NUNIT)=IREC
      END IF
      N=N+1
      GO TO 110
150 IF(N.EQ.0)           GO TO 10          00000780
      WRITE(NUNIT,REC=1) (ICONT(R(I,NUNIT),I=1,LCONT))
      WRITE(6,6240)              00000790
      WRITE(6,6120)              00000800
      WRITE(6,6125) (ICONT(R(LP1,NUNIT),LP1=1,18) 00000810
      WRITE(6,6130) ICONT(R(21,NUNIT)            00000820
      WRITE(6,6135) ICONT(R(22,NUNIT)            00000830
      WRITE(6,6140) ICONT(R(23,NUNIT)            00000840
      WRITE(6,6145) ICONT(R(24,NUNIT)            00000850
      WRITE(6,6150) ICONT(R(25,NUNIT)            00000860
      WRITE(6,6155) ICONT(R(26,NUNIT)            00000870
      WRITE(6,6160) ICONT(R(27,NUNIT)            00000880
      WRITE(6,6165) ICONT(R(28,NUNIT)            00000890
      WRITE(6,6170) ICONT(R(29,NUNIT)            00000900
      WRITE(6,6175) ICONT(R(30,NUNIT)            00000910
      WRITE(6,6180) ICONT(R(31,NUNIT)            00000920
      GO TO 10                      00000930
C MEND OF DIRECTORY SECTION          00000940
200 WRITE(6,6300)                  00000950
210 CALL NODINP(NTH, NODE)          00000960
      IF(NTH.EQ.0)           GO TO 10          00000970
      CALL PFIND(NUNIT, NODE, NTH, NDIRC, LLL)
      IF(LLL.EQ.0)           GO TO 220         00000980
      WRITE(6,6310)              00000990
      GO TO 210                  00001000
220 WRITE(6,6320)                  00001010
230 READ(5,5000) REC               00001020
      IF(REC(1:3).EQ.'SUB')    GO TO 250         00001030
      IF(REC(1:4).EQ.'HEAD')    GO TO 410         00001040
      WRITE(6,6110)              00001050
      GO TO 230                  00001060
C MEND SUB-DIRECTORY          00001070
250 NUPD=0                      00001080
C SEARCH SUB-DIRECTORY NO.          00001090
  ITEM=IBUFFR(4)
  DO 260 N=1,ITEM
    IF(IBUFFR(5+12*(N-1)).EQ.NODE(NTH)) THEN
      NSUB=N
      GO TO 270
    END IF
260 CONTINUE                      00001100
270 CALL PRSUBD(NDIRC)            00001110
275 WRITE(6,6330)                  00001120
280 READ(5,5000) REC               00001130
      IF(REC(1:3).EQ.'DEL')    GO TO 290         00001140
      KC=ICCDNV(REC)
      IF(KC.EQ.0)           GO TO 400         00001150

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0009

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0010

```

        IF(KC.GE.1 .AND. KC.LE.12)          GO TO 320          00001250
        WRITE(6,6210)
        GO TO 280
C      DELETE SUB-DIRECTORY
290  IF(NSUB.EQ.1ITEM)                 GO TO 310          00001260
        NN1=5+12*(NSUB-1)
        NN2=NN1+12
        DO 300 N=NSUB+1,ITEM
        DO 300 I=1,12
            IBUFFR(NN1)=IBUFFR(NN2)
            NN1=NN1+1
300   NN2=NN2+1
310  IBUFFR(4)=ITEM-1
        WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
        WRITE(6,6340) NODE(NTH)
        GO TO 10
C      UPDATE SUB-DIRECTORY
320  WRITE(6,6220)
        READ(5,5000) REC
        NUPD=NUPD+1,
        NADD=4+12*(NSUB-1)
        IF(KC.EQ.1) THEN
            READ(REC,'(A4)') IBUFFR(NADD+KC)
            GO TO 275
        END IF
        IF(KC.GE.2 .AND. KC.LE.5) THEN
            IREC=ICCONV(REC)
            IBUFFR(NADD+KC)=IREC
            GO TO 275
        END IF
        IF(KC.EQ.6) THEN
            READ(REC,'(2A4)') IBUFFR(NADD+KC),IBUFFR(NADD+KC+1)
            GO TO 275
        END IF
        IF(KC.GE.8) THEN
            CALL CVDAT(REC,ITYP,1DAT,RDAT,CDAT)
            GO TO (330,340,350),ITYP
330   IBUFFR(NADD+KC)=1DAT
            WRITE(6,6350) 1DAT
            GO TO 275
340   RBUFFR(NADD+KC)=RDAT
            WRITE(6,6360) RDAT
            GO TO 275
350   READ(CDAT,'(A4)') IBUFFR(NADD+KC)
            WRITE(6,6370) CDAT
            GO TO 275
        END IF
400  IF(NUPD.EQ.0)                      GO TO 10          00001500
        WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
        WRITE(6,6380)
        CALL PRSUBD(IBUFFR(NADD+1))
        GO TO 10
C      MEND DIRECTORY HEAD
410  NUPD=0
        IX=NDIRC(3)          00001510
                                00001520
                                00001530
                                00001540
                                00001550
                                00001560
                                00001570
                                00001580
                                00001590
                                00001600
                                00001610
                                00001620
                                00001630
                                00001640
                                00001650
                                00001660
                                00001670
                                00001680
                                00001690
                                00001700
                                00001710
                                00001720
                                00001730
                                00001740
                                00001750
                                00001760
                                00001770
                                00001780
                                00001790

```

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0010

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0011

```

        IF(IX.EQ.0) THEN          00001800
          WRITE(6,6400)           00001810
          GO TO 10                00001820
        END IF                   00001830
        READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00001840
        WRITE(6,6410) (IBUFFR(I),I=1,4)           00001850
        ITEM=IBUFFR(4)             00001860
        WRITE(6,6420) (IBUFFR(5+12*(I-1)),I=1,ITEM) 00001870
420  WRITE(6,6430)             00001880
430  READ(5,5000) REC          00001890
        IF(REC(1:3).EQ.'DEL')    GO TO 440 00001900
        KC=1CCONV(REC)           00001910
        IF(KC.EQ.0)               GO TO 460 00001920
        IF(KC.GE.1 .AND. KC.LE.4) GO TO 450 00001930
        WRITE(6,6210)              00001940
        GO TO 430                00001950
C   DELETE DIRECTORY HEAD      00001960
440  READ(SLASH,'(A4)') IBUFFR(1) 00001970
        IBUFFR(4)=0               00001980
        WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00001990
        WRITE(6,6440) NODE(NTH)       00002000
        GO TO 10                  00002010
C   UPDATE DIRECTORY HEAD      00002020
450  WRITE(6,6220)              00002030
        READ(5,5000) REC          00002040
        NUPD=NUPD+1               00002050
        IF(KC.LE.2) THEN          00002060
          READ(REC,'(A4)') IBUFFR(KC)
          GO TO 420                00002070
        END IF                   00002080
        IF(KC.EQ.3 .OR. KC.EQ.4) THEN 00002090
          IREC=1CCONV(REC)
          IBUFFR(KC)=IREC
          GO TO 420                00002100
        END IF                   00002110
460  IF(NUPD.EQ.0)             GO TO 10 00002120
        WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00002130
        WRITE(6,6450)              00002140
        WRITE(6,6410) (IBUFFR(I),I=1,4)           00002150
        GO TO 10                  00002160
C   END OF COMMENT            00002170
500  WRITE(6,6300)              00002180
510  CALL NODINP(NTH,NODE)      00002190
        IF(NTH.EQ.0)               GO TO 10 00002200
        CALL PFIND(NUNIT,NODE,NTH,NOIRC,LLL) 00002210
        IF(LL.NE.0) THEN          00002220
          WRITE(6,6310)              00002230
          GO TO 510                00002240
        END IF                   00002250
        NDASET=NDIRC(5)           00002260
        IF(NDASET.EQ.0) THEN      00002270
          WRITE(6,6500)              00002280
          GO TO 510                00002290
        END IF                   00002300
        WRITE(6,6510) NDASET       00002310
                                            00002320
                                            00002330
                                            00002340

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0011

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0012

```

DO 540 N=1,NDASET                         00002350
NDAIX(N)=IX                               00002360
CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASUDD,NOSUBD,NOARY,NDATA) 00002370
NWORD=25+NOARY                           00002380
DO 520 I=1,NOARY                          00002390
520 NWORD=NWORD+NDATA(I)                  00002400
NRECD=NWORD/LRECOD                         00002410
IF(HOD(NWORD,LRECOD).EQ.0)                GO TO 530 00002420
NRECO=NRECD+1                            00002430
530 IX=IX+NRECO                           00002440
540 WRITE(6,6520) N,(ICH(I),I=1,18)       00002450
550 WRITE(6,6530)                           00002460
560 READ(5,5000) REC                      00002470
IREC=ICCONV(REC)                         00002480
IF(IREC.EQ.0)                            GO TO 590 00002490
IF(IREC.GE.1 .AND. IREC.LE.NDASET)      GO TO 570 00002500
WRITE(6,6540)                            00002510
GO TO 560                                00002520
570 WRITE(6,6550)                           00002530
READ(5,5100) (ICH(I),I=1,20)             00002540
IX=NDAIX(IREC)                           00002550
READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00002560
DO 580 I=1,20                             00002570
580 IBUFFR(I+2)=ICH(I)                   00002580
WRITE(NUNIT,REC=IX)(IBUFFR(I),I=1,LRECOD) 00002590
GO TO 550                                00002600
590 WRITE(6,6560)                           00002610
DO 600 N=1,NDASET                         00002620
IX=NDAIX(N)                             00002630
CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASU8D,NOSUBD,NOARY,NDATA) 00002640
600 WRITE(6,6520) N,(ICH(I),I=1,18)       00002650
GO TO 10                                 00002660
C END OF MEND                           00002670
1000 WRITE(6,6600)                         00002680
ICONTR(24,NUNIT)=0                        00002690
WRITE(NUNIT,REC=1) (ICONTR(I,NUNIT),I=1,LCTR) 00002700
2000 STOP                                00002710
5000 FORHAT(A80)                          00002720
5100 FORHAT(20A4)                         00002730
6000 FORMAT(' ++ DATA POOL INFORMATION ++',
+        '/ TITLE : /1X,18A4                 00002750
+        '/ 1ST. RECORD NO. OF DIRECTORY : ,15, 00002760
+        '/ LAST RECORD NO. OF DIRECTORY : ,15, 00002770
+        '/ REAL RECORD NO. OF DIRECTORY : ,15, 00002780
+        '/ WRITE PERMIT OF THE DATA POOL : ,15) 00002790
6010 FORMAT(1H , DATA POOL ACCESS IS INHIBITED.,/, CHECK ALL OTHER 00002800
*ACCESS OF THE DATA POOL',/, IF THE DATA POOL ACCESS IS NOW ALLOW00002810
+ED',/, ENTER OK SIGN FROM YOUR TERMINAL') 00002820
6100 FORMAT(' ENTER OPTION NAME CONT/DIREC/COM/END') 00002830
6110 FORMAT(' WRONG OPTION NAME. PLEASE RE-ENTER OPTION NAME') 00002840
6120 FORMAT(1H0,'***** C O N T R O L S E C T I O N *****'00002850
*     /1H0,5X,'ITEM')
6125 FORMAT(1H0,6X,'1 TITLE : /1H0,6X,18A4) 00002860
6130 FORMAT(1H0,6X,'21 ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE 00002880
*: ',1I0)                                00002890

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16 (FRIDAY) TIME 16:52:25 PAGE 0012

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0013

6135	FORMAT(1H0,6X,'22 HEAD ADDRESS FOR THE VACANT DIRECTORY AREA *: ',110)	00002900 00002910
6140	FORMAT(1H0,6X,'23 HEAD ADDRESS FOR THE VACANT DATA AREA *: ',110)	00002920 00002930
6145	FORMAT(1H0,6X,'24 WRITE FLAG *: ',110)	00002940 00002950
6150	FORMAT(1H0,6X,'25 READ FLAG (NOT USED) *: ',110)	00002960 00002970
6155	FORMAT(1H0,6X,'26 LENGTH OF THE ONE PHYSICAL RECORD *: ',110)	00002980 00002990
6160	FORMAT(1H0,6X,'27 MAXIMUM NUMBER OF THE SAME LEVEL NODE *: ',110)	00003000 00003010
6165	FORMAT(1H0,6X,'28 SIZE OF THE DIRECTORY SECTION *: ',110)	00003020 00003030
6170	FORMAT(1H0,6X,'29 SIZE OF THE DATA SECTION *: ',110)	00003040 00003050
6175	FORMAT(1H0,6X,'30 REAL NUMBER OF THE DIRECTORY RECORDS *: ',110)	00003060 00003070
6180	FORMAT(1H0,6X,'31 REAL NUMBER OF THE DATA SET RECORDS *: ',110)	00003080 00003090
6200	FORMAT(' ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS')	00003100
6210	FORMAT(' WRONG ITEM NO. PLEASE RE-ENTER ITEM NO.')	00003110
6215	FORMAT(' ENTER NEW TITLE')	00003120
6220	FORMAT(' ENTER NEW VALUE')	00003130
6230	FORMAT(' WRONG VALUE WAS ENTERED. PLEASE RE-ENTER ITEM NO.')	00003140
6240	FORMAT(' END OF MENDING A CONTROL SECTION SUCCESSFULLY')	00003150
6300	FORMAT(' ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS')	00003160
6310	FORMAT(' NOT FOUND NODE NAME. PLEASE RE-ENTER NODE NAME')	00003170
6320	FORMAT(' ENTER OPTION NAME SUB/HEAD')	00003180
6330	FORMAT(' ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTORY') *Y/*' IF ENTER 0, END TO MEND THE SUB-DIRECTORY')	00003190 00003200
6340	FORMAT(' SUB-DIRECTORY ',A4,' WAS DELETED')	00003210
6350	FORMAT(' INPUT VALUE WAS INTEGER TYPE ',I5)	00003220
6360	FORMAT(' INPUT VALUE WAS REAL TYPE ',1PE12.3)	00003230
6370	FORMAT(' INPUT VALUE WAS CHARACTER TYPE ',A4)	00003240
6380	FORMAT(' END OF MENDING A SUB-DIRECTORY SECTION SUCCESSFULLY')	00003250
6400	FORMAT(' DIRECTORY HEAD DID NOT EXIST.')	00003260
6410	FORMAT(' DIRECTORY HEAD/' 1 ' ITEM CONTENTS '/ 2 ' 1 NODE NAME 1 ',1X,A4/ 3 ' 2 NODE NAME 2 ',1X,A4/ 4 ' 3 UPPER DIRECTORY ADDRESS ',15/ 5 ' 4 NO. OF SUB-DIRECTORY ',15)	00003270 00003280 00003290 00003300 00003310 00003320
6420	FORMAT(' NODE NAMES FOR EACH SUB-DIRECTORY'/(12(2X,A4)))	00003330
6430	FORMAT(' ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS DIRECTORY HE00003340 *AD'/' IF ENTER 0, END TO MEND THE DIRECTORY')	00003350
6440	FORMAT(' DIRECTORY HEAD ',A4,' WAS DELETED')	00003360
6450	FORMAT(' END OF MENDING A DIRECTORY SECTION SUCCESSFULLY')	00003370
6500	FORMAT(' THERE IS NO DATA RECORD. PLEASE RE-ENTER NODE NAME')	00003380
6510	FORMAT(' NO. OF SUB-DATA-SET IS ',I3/ * ' DAT NO. COMMENT')	00003390 00003400
6520	FORMAT(I7,1X,18A4)	00003410
6530	FORMAT(' ENTER DAT NO. TO MEND. IF ENTER 0, END TO PROCESS')	00003420
6540	FORMAT(' WRONG DAT NO. PLEASE RE-ENTER DAT NO.')	00003430
6550	FORMAT(' ENTER NEW COMMENT')	00003440

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0013

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0014

```

6560 FORMAT(' END OF MENDING DATA COMMENTS SUCCESSFULLY'
           *      ' DAT NO.      COMMENT')
6600 FORMAT(' END OF MEND COMMAND')
END

```

```

*****
** MTCOPY   **
*****

```

```

C*****00000100
C*          *00000200
C*          MTCOPY      *00000300
C*          *00000400
C*****00000500
PROGRAM      MTCOPY      00000600
DIMENSION    JCONTR(40),IDATE(2) 00000700
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000800
C          00000900
COMMON /DPWORK/ IBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00001000
*          NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD, 00001100
*          NTHOLD,NODOLD(10,2),NA1 00001200
CHARACTER*4 BLANK      00001300
DATA BLANK/'  '/ 00001400
C          00001500
NUNIT=91      00001600
CALL POPEN(NUNIT,JCONTR) 00001700
C RESET CONTROL SECTION 00001800
REWIND 1      00001900
READ(1) IDATE,(ICONT(1,NUNIT),I=1,20) 00002000
ICONT(21,NUNIT)=2 00002100
ICONT(22,NUNIT)=3 00002200
ICONT(23,NUNIT)=ICONT(28,NUNIT)+2 00002300
ICONT(24,NUNIT)=1 00002400
ICONT(30,NUNIT)=1 00002500
ICONT(31,NUNIT)=0 00002600
NUTOLD=0      00002700
NTHOLD=0      00002800
NDSTAT=ICONT(23,NUNIT) 00002900
WRITE(NUNIT,REC=1) (ICONT(LP1,NUNIT),LP1=1,LCONTR) 00003000
READ(BLANK,'(A4)') IBUFFR(1) 00003100
READ(BLANK,'(A4)') IBUFFR(2) 00003200
IBUFFR(3)=0 00003300
IBUFFR(4)=0 00003400
WRITE(NUNIT,REC=2) (IBUFFR(LP1),LP1=1,4) 00003500
C LOAD BACK-UP DATA 00003600
CALL PRECVR(NUNIT,2,LLL) 00003700
WRITE(6,6000) 00003800
CALL PWEND(NUNIT) 00003900
C          00004000
1000 STOP      00004100
C          00004200
6000 FORMAT('0 MTCOPY WAS FINISHED SUCCESSFULLY') 00004300
END      00004400

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0014

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0015

```
*****
** MTSAVE **
*****
```

```

C*****00000010
C*          *00000020
C*          MTSAVE          *00000030
C*          *00000040
C*****00000050
C*          PROGRAM MTSAVE          00000060
C*          DIMENSION JCONTR(40),NODE(10),IDATE(2)          00000070
C*          COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT          00000080
C*          00000090
C*          COMMON /DPWORKS/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,          00000100
C*          *NAOWN,NADAT,NASET,NDATE(2),NINFOM(5),NUTOLD,          00000110
C*          *NTHOLD,NODOLD(10,2),NA1          00000120
C*          COMMON /DPCUNT/ NOREC,NONOD          00000130
C*          00000140
C*          NUNIT=91          00000150
C*          CALL POPEN(NUNIT,JCONTR)          00000160
C*          CALL DATE(IDATE)          00000170
C*          NREC=0          00000180
C*          NONOD=0          00000190
C*          REWIND 1          00000200
C*          WRITE(1) IDATE,(ICONTROL(I,NUNIT),I=1,20),(ICONTROL(I,NUNIT),I=26,31) 00000210
C*          00000220
C*          WRITE(6,6000)          00000240
C*          WRITE(6,6100) (ICONTROL(I,NUNIT),I=1,20)          00000250
C*          WRITE(6,6200) ICONTROL(26,NUNIT)          00000260
C*          WRITE(6,6300) ICONTROL(27,NUNIT)          00000270
C*          WRITE(6,6400) ICONTROL(28,NUNIT),ICONTROL(30,NUNIT)          00000280
C*          WRITE(6,6500) ICONTROL(29,NUNIT),ICONTROL(31,NUNIT)          00000290
C*          WRITE(6,6600) (N,N=1,8)          00000340
C*          IX=ICONTROL(21,NUNIT)          00000350
C*          NTH=0          00000360
C*          KEY=3          00000370
C*          CALL PNLIST(NUNIT,NODE,NTH,KEY,LLL)          00000380
C*          IF(LLL.EQ.1)          GO TO 900          00000390
C*          00000400
C*          WRITE(6,6700)          00000410
C*          WRITE(6,6800) NOREC,NONOD          00000411
C*          GO TO 1000          00000412
C*          900 WRITE(6,6900)          00000413
C*          1000 STOP          00000420
C*          6000 FORMAT(1H1////20X,'N O D E T R E E')          00000720
C*          6100 FORMAT(1H0,1X,'TITLE OF A DATA POOL      ***'//6X,20A4) 00000730
C*          6200 FORMAT(1H0,1X,'LENGTH OF A RECORD      *** ',5X,15) 00000740
C*          6300 FORMAT(1H0,1X,'MAXIMUM NUMBER OF THE SAME LEVEL NODE *** ',5X,15) 00000750
C*          6400 FORMAT(1H0,1X,'SIZE OF THE DIRECTORY SECTION      *** ',5X,15, 00000760
C*          *           3X,'(USED RECORDS',15,')')          00000770
C*          6500 FORMAT(1H0,1X,'SIZE OF THE DATA SECTION      *** ',5X,15, 00000780
C*          *           3X,'(USED RECORDS',15,')')          00000790
C*          6600 FORMAT(//1H,'LEVEL      ',8(I1,6X))          00000820
C*          6700 FORMAT('O  MTSAVE WAS FINISHED SUCCESSFULLY')          00000840

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0015

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0016

```

6800 FORMAT('O      NO. OF DIRECTORY = ',15/
           *          '      NO. OF NODE     = ',15)
6900 FORMAT('O  ABNORMAL END')
      END

```

\*\*\*\*\*  
\*\* RENAME \*\*  
\*\*\*\*\*

```

C*****00000100
C*00000200
C*00000300
C*00000400
C*****00000500
      PROGRAM  RENAME
      DIMENSION JCTR(40),NODEA(10),NODEB(10),NDIRC(12)
      COMMON /DPCONT/ LCTR,NCTR,ICTR(40,99),IX,NSUBDS,NSTAT
C      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
*                      NAOWN,NADAT,NDASET,NDATE(2),NINFOR(5),NUTOLD,
*                      NTHOLD,HOOLD(10,2),NA1
C      NUNIT=91
      CALL POPEN(NUNIT,JCTR)
10    WRITE(6,6000)
      CALL NODINP(NTH1,NODEA)
      IF(NTH1.EQ.0)               GO TO 1000
      WRITE(6,6010)
      CALL NODINP(NTH2,NODEB)
C      CHECK OF INPUT NODE
      IF(NTH1.EQ.NTH2)           GO TO 100
      WRITE(6,6020)
      GO TO 10
100   IF(NTH1.EQ.1)             GO TO 120
      DO 110 N=1,NTH1-1
      IF(NODEA(N).EQ.NODEB(N))
          WRITE(6,6030)
          GO TO 10
110   CONTINUE
120   IF(NODEA(NTH1).NE.NODEB(NTH1))  GO TO 130
      WRITE(6,6040)
      GO TO 10
130   CALL PFIND(NUNIT,NODEB,NTH2,NDIRC,LLL)
      IF(LLL.NE.0)                 GO TO 140
      WRITE(6,6045)
      GO TO 10
140   CALL PFIND(NUNIT,NODEA,NTH1,NDIRC,LLL)
      IF(LLL.EQ.0)                 GO TO 200
      WRITE(6,6050)
      GO TO 10
C      START RENAME
200   ITEM=IBUFFR(4)
      DO 210 N=1,ITEM
      IF(IBUFFR(5+12*(N-1)).EQ.NODEA(NTH1)) THEN
          NSUB=N

```

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0016

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0017

```

        GO TO 220                                00004700
        END IF                                 00004800
210 CONTINUE                                00004900
220 IBUFFR(5+12*(NSUB-1))=NODEB(NTH2)      00005000
        NADWN=IBUFFR(7+12*(NSUB-1))           00005100
        NADAT=IBUFFR(8+12*(NSUB-1))           00005200
        NDASET=IBUFFR(9+12*(NSUB-1))          00005300
        WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
C   CHANGE LOWER NODE DIRECTORY              00005400
        IF(NADWN.EQ.0)                         GO TO 250  00005500
        IX=NAOWN                               00005600
        READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00005700
        IBUFFR(1)=NODEB(NTH2)                  00005800
        WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00005900
C   CHANG DATA SECTION                     00006100
250 IF(NADAT.EQ.0)                         GO TO 300  00006200
        IX=NADAT                               00006300
        DO 270 N=1,NDASET                      00006400
        READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00006500
        IBUFFR(1)=NODEB(NTH2)                  00006600
        NOARY=IBUFFR(25)                      00006700
        WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD) 00006800
        NWORD=NWORD+1BUFFR(25+1)               00006900
        DO 260 I=1,NOARY                      00007000
        NWORD=NWORD+1BUFFR(25+1)               00007100
        NRECD=NWORD/LRECOD                   00007200
        IF(NODC(NWORD,LRECOD).EQ.0)            GO TO 270  00007300
        NRECD=NRECD+1
270 IX=IX+NRECD                           00007500
300 CONTINUE                                00007600
        WRITE(6,6060)                          00007700
        GO TO 10                                00007800
C
1000 STOP                                  00007900
C
6000 FORMAT(' ENTER A OLD NODE NAME')       00008200
6010 FORMAT(' ENTER A NEW NODE NAME')       00008300
6020 FORMAT(' THE LEVEL OF TWO NODES ARE DIFFERENT. PLEASE RE-ENTER') 00008400
6030 FORMAT(' A NODE NAME TO BE RENAMED IS ONLY LAST ONE. PLEASE RE-ENTER') 00008500
*ER')
6040 FORMAT(' A LAST NODE NAME IS SAME. PLEASE RE-ENTER') 00008700
6045 FORMAT(' NEW NODE NAME EXIST ALREADY. PLEASE RE-ENTER') 00008800
6050 FORMAT(' A OLD NODE NAME IS NOT FOUND. PLEASE RE-ENTER') 00008900
6060 FORMAT(' RENAME IS FINISHED SUCCESSFULLY') 00009000
END                                         00009100

```

```

*****
** TREE **
*****

```

```

*****C00000100
C*          *C00000200
C*          *C00000300
C*          *C00000400
*****C00000500

```

FILE = J3679.POOL2.FORT77 DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0017

FILE = J3679.POOL2.FORT77

DATE 84/03/16(FRIDAY) TIME 16:52:25 PAGE 0018

```

PROGRAM TREE                                00000600
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT      00000700
COMMON /DPWORK/ LBUFFR,LRECOD,IBUFR(1000),NRECOO,NODE1,NODE2,    00000800
1          NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,        00000900
1          NTOLD,NODOLD(10,2),NA1                           00001000
DIMENSION JCTR(40),NODE(10)                   00001100
NUNIT=91                                      00001200
CALL POPEN(NUNIT,JCTR)
WRITE(6,1000) (ICONT(1,NUNIT),I=1,20)        00001300
WRITE(6,1200) ICONT(26,NUNIT)                 00001400
WRITE(6,1300) ICONT(27,NUNIT)                 00001500
WRITE(6,1400) ICONT(28,NUNIT),ICONT(30,NUNIT)   00001600
WRITE(6,1500) ICONT(29,NUNIT),ICONT(31,NUNIT)   00001700
NN1=1+ICONT(28,NUNIT)-ICONT(22,NUNIT)+1       00001800
NN2=1+ICONT(28,NUNIT)+1+ICONT(29,NUNIT)-ICONT(23,NUNIT)+1   00001900
WRITE(6,1600) NN1
WRITE(6,1700) NN2
WRITE(6,1800) (N,N=1,8)                      00002000
IX=ICONT(21,NUNIT)                           00002100
NTH=0                                         00002200
KEY=2                                         00002300
CALL PHLST(NUNIT,NODE,NTH,KEY,LLL)           00002400
STOP                                         00002500
00002600
00002700
00002800
00002900
1000 FORMAT(1H1////20X,'N O D E T R E E')      00003000
1100 FORMAT(1HO,1X,'TITLE OF A DATA POOL      ***'//6X,20A4) 00003100
1200 FORMAT(1HO,1X,'LENGTH OF A RECORD        *** ',5X,I5) 00003200
1300 FORMAT(1HO,1X,'MAXIMUM NUMBER OF THE SAME LEVEL NODE *** ',5X,I5) 00003300
1400 FORMAT(1HO,1X,'SIZE OF THE DIRECTORY SECTION *** ',5X,I5) 00003400
*          3X,'USED RECORDS',15,'')
1500 FORMAT(1HO,1X,'SIZE OF THE DATA SECTION   *** ',5X,I5) 00003500
*          3X,'USED RECORDS',15,'')
1600 FORMAT(1HO,1X,'REMAINS OF THE DIRECTORY SECTION *** ',5X,I5) 00003600
1700 FORMAT(1HO,1X,'REMAINS OF THE DATA SECTION  *** ',5X,I5) 00003700
1800 FORMAT(//1H , 'LEVEL      ',8(11,6X))      00003800
1800 FORMAT(//1H , 'LEVEL      ',8(11,6X))      00003900
1800 FORMAT(//1H , 'LEVEL      ',8(11,6X))      00004000
END                                         00004100

```

## DIRECTORY LIST OF J3679.POOLC2.FORT??

***** MEMBER NAME *****	***** PAGE NO. *****	***** NO. OF CARDS *****
(NO.=001) CVOAT	0001	57
(NO.=002) ICCONV	0002	26
(NO.=003) NODINP	0002	28
(NO.=004) PHLST	0003	65
(NO.=005) PRECVR	0004	87
(NO.=006) PRSUBD	0006	56
(NO.=007) PSAVE	0007	91
(NO.=008) PWLIST	0009	34

-----  
444 CARDS

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

```
*****
** CVDAT   **
*****
```

```

      SUBROUTINE CVDAT(REC,ITYP,IDAT,RDAT,CDAT)          00000010
C                                         00000020
C   CONVERT INPUT REC TO EACH TYPE ACCORDING TO THE ATTRIBUTE 00000030
C                                         00000040
C
C   CHARACTER REC*80,CDAT*4,DIGIT(10)*1,FLOAT(2)*1,SIGN(2)*1,IBLK*1, 00000050
*           CHR*1,NUH*12,BLANK*12                           00000060
    DATA DIGIT,FLOAT,SIGN,IBLK/'0','1','2','3','4','5','6','7','8',
*           '9','.',',','E','+',','-','/'                  00000070
    DATA BLANK/' '
    ITYP=1
    N=0
    DO 200 NC=1,13
    CHR=REC(NC:NC)
    IF(CHR.EQ.IBLK .AND. N.EQ.0)          GO TO 200        00000140
    IF(CHR.EQ.IBLK .AND. N.GT.0)          GO TO 300        00000150
    N=N+1
C   SIGN                                         00000160
    DO 100 I=1,2
    IF(CHR.EQ.SIGN(I))          GO TO 200        00000170
100   CONTINUE                                     00000180
C   DIGIT                                         00000190
    DO 120 I=1,10
    IF(CHR.EQ.DIGIT(I))          GO TO 200        00000200
120   CONTINUE                                     00000210
C   FLOAT                                         00000220
    DO 140 I=1,2
    IF(CHR.EQ.FLOAT(I))          GO TO 160        00000230
140   CONTINUE                                     00000240
C   OTHERS                                         00000250
    ITYP=3
    GO TO 300                                     00000260
160   ITYP=2
200   CONTINUE                                     00000270
    GO TO 450                                     00000280
300   NE=NC-1                                     00000290
C
    IF(ITYP.EQ.3)          GO TO 400        00000300
    NUH=BLANK                                     00000310
    NS=12-NE+1                                     00000320
    NUM(NS:12)=REC(1:NE)                         00000330
    IF(ITYP.EQ.2)          GO TO 350        00000340
C   INTEGER                                         00000350
    READ(NUM,'(I12)')  IDAT
    GO TO 500                                     00000360
C   FLOATING                                         00000370
    350  READ(NUM,'(F12.0)')  RDAT
    GO TO 500                                     00000380
C   CHARACTER                                         00000390
    400  CDAT=REC(1:4)                           00000400
    GO TO 500                                     00000410

```

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

```

C ERROR
 450 WRITE(6,6000)
    IDAT=0
    500 RETURN
C
 6000 FORMAT(' INPUT DATA ERROR. DATA MUST BE IN COL. 1 - 12')
    END
*****  

** ICCONV **  

*****  

C-----+-----C 00000010
C++++++ FUNCTION ICCONV ++++++ C 00000020
C-----+-----C 00000030
C
  FUNCTION ICCONV(N)
  CHARACTER*80 N
  CHARACTER*1 J(11)
  DATA J/'1','2','3','4','5','6','7','8','9','0',' '/
C
  MH=0
  DO 2 K=1,72
  DO 3 M=1,11
  IF(J(M).EQ.N(K)) GO TO 4
  3 CONTINUE
  GO TO 5
  4 CONTINUE
  IF(N.EQ.10) MH=0
  IF(N.EQ.11) GO TO 2
  MH=MH*10+M
  2 CONTINUE
  ICCONV=MH
  RETURN
  5 WRITE(6,6) N(K:K),K
  ICCONV=-1
  RETURN
  6 FORMAT(1H , ' +IRREGULAR CHARACTOR ',A1,' IN COL.',I3)
  END
*****  

** NODINP **  

*****  

C
  SUBROUTINE NODINP(NTH,NODE)
C
  READ NODE NAME LIST
C
  CHARACTER*4 NODE(10)
  CHARACTER BUFF*80
C
  10 READ(5,6000,END=1000) BUFF
  N=0
  NTH=0
  100 N=N+1

```

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

```

      IF(BUFF(N:N+3).EQ.'      ')      GO TO 200      00001300
      NTH=NTH+1                      00001400
      NODE(NTH)=BUFF(N:N+3)          00001500
      N=N+4                          00001600
      IF(BUFF(N:N).EQ.'.')           GO TO 100      00001700
      IF(BUFF(N:N).EQ.' ')           GO TO 200      00001800
      WRITE(6,6010)                  00001900
      GO TO 10                        00002000
      200 RETURN                      00002100
      1000 NTH=0                      00002200
      RETURN                          00002300
      6000 FORMAT(A80)                00002400
      6010 FORMAT(' WRONG NODE NAME. INPUT NODE NAME MUST FOLLOW THE TYPE SH000002500
      *WN BELOW'/'      'AAAA.BBBB.CCCC.DDDD ...'/' PLEASE RE-ENTER0002600
      * NODE NAME')                  00002700
      END                            00002800

*****
** PNLST  **
*****
```

```

C
      SUBROUTINE PNLST(NUNIT,NODE,NTH,KEY,LLL)      00000100
C
      C SEARCH LOWER NODE NEST                      00000200
      C   KEY=1 PUT DIRECTORY AND DATA TO A SEQUENTIAL FILE (ITP1) 00000300
      C           AND NODE NAMES TO A FILE (ITP2)                   00000400
      C   KEY=2 PRINT OF NODE NAME TREE               00000500
      C   KEY=3 BOTH                                00000600
      C
      C DIMENSION NODE(10)                         00000700
      C COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT 00000800
      C
      C COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00000900
      *          NAOWN,NADAT,NASET,NDATE(2),NINFOM(5),NUTOLD,        00001000
      *          NTIOLD,NDOLD(10,2),NA1                         00001100
      C COMMON /DPCUNT/ NDREC,NOHOD                 00001200
      C DIMENSION ITEM(10),LL(10)                  00001300
      C LLL=0                                      00001400
      C NTH0=NTH                                     00001500
      C LVL=1                                      00001600
      C NTH=NTH+LVL                                 00001700
      C DO 10 N=1,10                                00001800
      C 10 LL(N)=1                                  00001900
      C
      C 100 READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)    00002000
      C     IF(LL(LVL).GT.1)                           GO TO 110    00002100
      C     ITEM(LVL)=IBUFFR(4)                         00002200
      C 110 IF(LL(LVL).GT.ITEM(LVL))                 GO TO 250    00002300
      C 120 NADD=4+12*(LL(LVL)-1)                    00002400
      C     NODE(NTH)=IBUFFR(NADD+1)                  00002500
      C     IX=IBUFFR(NADD+4)                         00002600
      C     IF(IBUFFR(NADD+5).EQ.0) IX=0             00002700
      C     IF(KEY.EQ.1 .OR. KEY.EQ.3) THEN          00002800
      C       CALL PSAVE(NUNIT,NODE,NTH,IBUFFR(NADD+1),NRETUN) 00002900
      C
```

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

```

        IF(NRETUN.NE.0)          GO TO 3100      00003500
        END IF                   00003600
        IF(KEY.EQ.2 .OR. KEY.EQ.3) THEN      00003700
          CALL PWLIST(NUNIT,NODE,NTH)      00003800
        END IF                   00003900
C     SET LOWER LEVEL NODE           00004000
        IX=IBUFFR(NADD+3)            00004100
        LL(LVL)=LL(LVL)+1          00004200
        IF(IX.EQ.0)                 GO TO 200      00004300
C     GO TO PROCESS OF LOWER NODE      00004400
        LVL=LVL+1                  00004500
        NTH=NTH+1                  00004600
        IF(LVL.GT.10)               GO TO 3000     00004700
        NDREC=NDREC+1              00004800
        GO TO 100                  00004900
C     PROCESS OF A NEXT SAME LEVEL NODE 00005000
        200 IF(LL(LVL).LE.ITEM(LVL))    GO TO 120      00005100
C     RETURN TO UPPER NODE          00005200
        250 LL(LVL)=1                00005300
          LVL=LVL-1                00005400
          NTH=NTH-1                00005500
          IF(LVL.EQ.0)               GO TO 300      00005600
          IX=IBUFFR(3)              00005700
          GO TO 100                  00005800
        300 NTH=NTH0                00005900
          RETURN                    00006000
        3000 WRITE(6,6000)           00006100
        3100 LLL=1                  00006200
          RETURN                    00006300
        6000 FORMAT(' THE DEPTH OF NODE LEVEL IS TOO LARGE. MAX IS 10') 00006400
        END                         00006500

```

\*\*\*\*\*  
\*\* PRECVR \*\*  
\*\*\*\*\*

```

SUBROUTINE PRECVR(IUNIT,KEY,LLL)      00000100
C
C     LOAD NODE DATA FROB BACK-UP FILE TO DATA POOL      00000200
C       KEY=1  IF SAME NODE NAME EXIST, ERROR RETURN      00000300
C       2  IF SAME NODE NAME EXIST, NOT OUTPUT NODE DATA AND 00000400
C           CONTINUE THE PROCESS                          00000500
C
C     DIMENSION INFOH(5),ICM(20)                         00000600
C     DIMENSION NODE(10),NDATA(4)                         00000700
C     COMMON /PPP/ DATA(50000)                           00000800
C     COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NSTAT 00000900
C
C     COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2, 00001000
C                   * NADWH,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD, 00001100
C                   * NTHOLD,NOOOLD(10,2),NA1                00001200
C     COMMON /DPCUNT/ NDREC,NONOD                         00001300
C
C     LLL=0                                              00001400
10 READ(1,END=700) N,(NODE(1),I=1,N),NADAT,NDASET,NDATE,INFOH 00001500
    CALL PSET(IUNIT,NODE,N,INFOH,0,NRETUN)                00001600

```

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

```

      IF(NRETUN.NE.0 .AND. KEY.EQ.1) GO TO 3000          00002000
      IF(NRETUN.NE.0 .AND. KEY.EQ.2) GO TO 550          00002100
      IF(HDASET.EQ.0) GO TO 600                         00002200
      DO 500 I=1,HDASET                                00002300
      READ(1) ICM,NTOT,NOARY,(NDATA(J),J=1,NOARY),
      1           (DATA(J),J=1,NTOT)                      00002400
      IF(NTOT.GT.50000)        GO TO 3100                00002500
      NOARY1=NOARY+1                                     00002600
      GO TO ( 50,100,200,300,400 ),NOARY1               00002700
      GO TO 3200                                         00002800
      50 CALL PRITE(IUNIT,ICM)                           00002900
      GO TO 500                                         00003000
      C
      CCC *** PRITE1 ROUTINE                          00003100
      C
      100 NDATA1 = 1                                    00003200
      CALL PRITE1(IUNIT,ICM,NDATA(1),DATA(NDATA1))     00003300
      GO TO 500                                         00003400
      C
      CCC *** PRITE2 ROUTINE                          00003500
      C
      200 NDATA1 = 1                                    00003600
      NDATA2 = NDATA1 + NDATA(1)                        00003700
      CALL PRITE2(IUNIT,ICM,NDATA(1),DATA(NDATA1),NDATA(2),DATA(NDATA2)) 00003800
      GO TO 500                                         00003900
      C
      CCC *** PRITE3 ROUTINE                          00004000
      C
      300 NDATA1 = 1                                    00004100
      NDATA2 = NDATA1 + NDATA(1)                        00004200
      NDATA3 = NDATA2 + NDATA(2)                        00004300
      CALL PRITE3(IUNIT,ICM,NDATA(1),DATA(NDATA1),NDATA(2),
      1           DATA(NDATA2),NDATA(3),DATA(NDATA3))   00004400
      GO TO 500                                         00004500
      C
      CCC *** PRITE4 ROUTINE                          00004600
      C
      400 NDATA1 = 1                                    00004700
      NDATA2 = NDATA1 + NDATA(1)                        00004800
      NDATA3 = NDATA2 + NDATA(2)                        00004900
      NDATA4 = NDATA3 + NDATA(3)                        00005000
      CALL PRITE4(IUNIT,ICM,NDATA(1),DATA(NDATA1),
      1           NDATA(2),DATA(NDATA2),
      1           NDATA(3),DATA(NDATA3),
      1           NDATA(4),DATA(NDATA4))                 00005100
      500 CONTINUE                                     00005200
      GO TO 600                                         00005300
      550 IF(HDASET.EQ.0)        GO TO 600              00005400
      DO 560 I=1,HDASET                                00005500
      560 READ(1)                                       00005600
      600 CONTINUE                                     00005700
      GO TO 10                                         00005800
      700 CONTINUE                                     00005900
      RETURN                                         00006000
      3000 WRITE(6,6000) (NODE(I),I=1,N)

```

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

```

      GO TO 3500                                00007500
3100  WRITE(6,6100) NTOT,(NODE(I),I=1,N)      00007600
      GO TO 3500                                00007700
3200  WRITE(6,6200) NOARY,(NODE(I),I=1,N)      00007800
3500  LLL=1                                    00007900
      RETURN                                     00008000
6000  FORMAT(' SAME NODE NAME EXIST IN A DATA POOL. THIS PROCESS IS ABEN00008100
      *D'      /'     NODE NAME = ',A4,9('.,A4))    00008200
6100  FORMAT(' EXCEED LENGTH OF DATA.'
      *      /'     LENGTH = ',I6,' NODE NAME = ',A4,('.,A4)) 00008300
6200  FORMAT(' EXCEED NO. OF ARRAY.'
      *      /'     NO. OF ARRAY = ',I3,' NODE NAME = ',A4,('.,A4)) 00008400
      END                                         00008500
      00008600
      00008700

*****
** PRSUBD  **
*****
```

```

      SUBROUTINE PRSUBD(NDIRC)                      00000100
C
C   PRINT OF SUB-DIRECTORY                         00000200
C
C   DIMENSION NDIRC(12)                            00000300
C   DIMENSION IFMAT(4)                            00000400
C   CHARACTER IIFMAT*8,IIFMAT*8,IRFMAT*8,IAFMAT*8,NCHR*1,MCHR*43,
      *          IBLANK*4,WORD*4                     00000500
C
C   DATA IFMAT(1),IFMAT(2),IFMAT(3)                00000600
      *          /'(1H ,I5,','6H  DATA','12,3H = '/      00000700
      *          I14)''                                00000800
      DATA IIFMAT/' 1PE14.4'''                      00000900
      DATA IRFMAT/'1PE14.4'''                      00001000
      DATA IAFMAT/'10X, A4)'''                      00001100
      DATA IAFMAT/'10X, A4)'''                      00001200
      DATA IBLANK/'      ''                          00001300
      DATA MCHR/'ABCDEFIGHJKLNMOPQRSTUVWXYZ0123456789* +-=( )'/
      DATA NOCHR/43/                                 00001400
      00001500
      00001600
      00001700
      00001800
      00001900
      00002000
      00002100
      00002200
      00002300
      00002400
      00002500
      00002600
      00002700
      00002800
      00002900
      00003000
      00003100
      00003200
      00003300
      00003400
      00003500
      00003600
      00003700

C   IMAX=2**30
C   IF(NDIRC(2).EQ.1BLANK) NDIRC(2)=0
      WRITE(WORD,'(A4)') NDIRC(2)
      IF(WORD.EQ.1BLANK) NDIRC(2)=0
      WRITE(6,6000) (NDIRC(LP3),LP3=1,7)
      DO 200 NC=8,12
      NN=NC-7
C   TEST IF CHARACTER
      WRITE(NCHR,'(A1)') NDIRC(NC)
      DO 100 N=1,NOCHR
      IF(NCHR.EQ.MCHR(N:N))      GO TO 110
      100  CONTINUE
      GO TO 120
      110  IFMAT(4)=IAFMAT
      120  IWRK=NDIRC(NC)
      IWRK=IABS(IWRK)
C   TEST IF INTEGER
      IF(IWRK.GE.IMAX)      GO TO 130
      0000200
      00002100
      00002200
      00002300
      00002400
      00002500
      00002600
      00002700
      00002800
      00002900
      00003000
      00003100
      00003200
      00003300
      00003400
      00003500
      00003600
      00003700
```

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

```

        IFMAT(4)=IIFMAT                               00003800
                                                GO TO 190 00003900
130      IFMAT(4)=IRFMAT                           00004000
190      CONTINUE                                 00004100
        WRITE(6,IFMAT) NC,NN,NDIRC(NC)             00004200
200      CONTINUE                                 00004300
C
C----- FORMAT
C
6000  FORMAT(' ITEM      CONTENTS'
*      1' 1 NODE NAME          =',4X,A4 00004400
*      1' 2 TOTAL LENG. OF DATA SET =',18 00004500
*      1' 3 ADDRESS OF A LOWER NODE =',18 00004600
*      1' 4 ADDRESS OF A DATA SET =',18 00004700
*      1' 5 NO. OF SUB-DATA SETS =',18 00004800
*      1' 6 DATE OF CREATION    =',2A4) 00004900
C
C      RETURN
END

*****
** PSAVE   **
*****


SUBROUTINE PSAVE(NUNIT,NODE,NTH,NDIRC,LLL) 00000100
C
C      SAVE NODE DATA TO A SEQUENTIAL FILE 00000200
C
C      DIMENSION NODE(1),NDIRC(1) 00000300
COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000400
COMMON /DPCUNT/ NDREC,NONOD 00000500
COMMON /PPP/ DATA(50000) 00000600
DIMENSION NDATA(4),ICM(20) 00000700
C
C
LLL=0                                         00000800
IXOLD=IX                                       00000900
NTOP = 50000                                     00001000
WRITE(1) NTH,(NODE(I),I=1,NTH),(NDIRC(I),I=4,12) 00001100
WRITE(2) NTH,(NODE(I),I=1,NTH)                 00001200
NONOD=NONOD+1                                    00001300
IF(NDIRC(5).EQ.0)                                GO TO 1000 00001400
DO 500 N=1,NDIRC(5)                            00001500
CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASDD,NOSDDS,NOARY,NDATA) 00001600
C
C
NOARY1=NOARY+1                                  00001700
GO TO (50,100,200,300,400),NOARY1            00001800
GO TO 3000                                       00001900
C
C
50 NDATA(1) = 0                                00002000
DATA(1)=0.0                                      00002100
WRITE(1) ICH,NDATA(1),NOARY,(NDATA(I),I=1,NOARY), 00002200
1           (DATA(I),I=1,NDATA(1))              00002300
GO TO 500                                       00002400
C
CCC *** PREAD1 ROUTINE                         00002500
                                              00002600
                                              00002700
                                              00002800
                                              00002900
                                              00003000
                                              00003100

```

FILE = J3679.POOLC2.FORT77 DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

```

C                               00003200
100 LAST    = NDATA(1)          00003300
     IF(LAST.GT.NTOP) GO TO 3100 00003400
     CALL PREAD1(NUNIT,ICH,N1,DATA)
     WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1           (DATA(I),I=1,LAST) 00003500
     GO TO 500                  00003600
C                               00003700
CCC *** PREAD2 ROUTINE      00003800
C                               00003900
200 NDATA1 = 1                00004000
     NDATA2 = NDATA1 + NDATA(1) 00004100
     LAST   = NDATA(1) + NDATA(2) 00004200
     IF(LAST.GT.NTOP) GO TO 3100 00004300
     CALL PREAD2(NUNIT,ICH,N1,DATA(NDATA1),N2,DATA(NDATA2))
     WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1           (DATA(I),I=1,LAST) 00004400
     GO TO 500                  00004500
C                               00004600
CCC *** PREAD3 ROUTINE      00004700
C                               00004800
300 NDATA1 = 1                00004900
     NDATA2 = NDATA1 + NDATA(1) 00005000
     NDATA3 = NDATA2 + NDATA(2) 00005100
     LAST   = NDATA(1) + NDATA(2) + NDATA(3) 00005200
     IF(LAST.GT.NTOP) GO TO 3100 00005300
     CALL PREAD3(NUNIT,ICH,N1,DATA(NDATA1),
     1           N2,DATA(NDATA2),
     2           N3,DATA(NDATA3))
     WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1           (DATA(I),I=1,LAST) 00005400
     GO TO 500                  00005500
C                               00005600
CCC *** PREAD4 ROUTINE      00005700
C                               00005800
400 NDATA1 = 1                00005900
     NDATA2 = NDATA1 + NDATA(1) 00006000
     NDATA3 = NDATA2 + NDATA(2) 00006100
     NDATA4 = NDATA3 + NDATA(3) 00006200
     LAST   = NDATA(1) + NDATA(2) + NDATA(3) + NDATA(4) 00006300
     IF(LAST.GT.NTOP) GO TO 3100 00006400
     CALL PREAD4(NUNIT,ICH,N1,DATA(NDATA1),
     1           N2,DATA(NDATA2),
     2           N3,DATA(NDATA3),
     3           N4,DATA(NDATA4))
     WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1           (DATA(I),I=1,LAST) 00006500
     GO TO 500                  00006600
C                               00006700
500 CONTINUE                  00006800
     IX=IXOLD                 00006900
1000 RETURN                   00007000
3000 WRITE(6,6000) NOARY,(NODE(I),I=1,NTH) 00007100
     GO TO 3500                 00007200
3100 WRITE(6,6100) LAST,(NODE(I),I=1,NTH) 00007300
3500 LLL=1                     00007400
     RETURN                    00007500
                                00007600
                                00007700
                                00007800
                                00007900
                                00008000
                                00008100
                                00008200
                                00008300
                                00008400
                                00008500
                                00008600

```

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY)

TIME 19:37:15 PAGE 0

FILE = J3679.POOLC2.FORT77

DATE 84/03/14(WEDNESDAY) TIME 19:37:15 PAGE 0

```

6000 FORMAT(' EXCEED NO. OF ARRAY.'
           *      // ' NO. OF ARRAY =',I3,' NODE NAME =',A4,9(.,A4))      00008700
6100 FORMAT(' EXCEED LENGTH OF DATA.'
           *      // ' LENGTH =',I6,' NODE NAME =',A4,9(.,A4))      00008800
           END                                              00008900
           00009000
           00009100

*****  

** PWLIST - **  

*****  

*****  

      SUBROUTINE PWLIST(NUNIT,NODE,NO)          00000100
C      C PRINT OF NODE TREE                   00000200
C      COMMON /DPCONT/ LCONTR,NCONTR,ICTR(40,99),IX,NSUBDS,NDSTAT 00000300
      DIMENSION NODE(10),NDATA(4)                00000400
      CHARACTER*4 ICH(20),BLANK                  00000500
      DATA BLANK//      ''                      00000600
C      IF(NO.GT.8)          GO TO 1000          00000700
      IF(IX.EQ.0) THEN                         00000800
         DO 10 N=1,20                           00000900
10     ICH(N)=BLANK
      ELSE
         CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA) 00001000
      ENDIF
      IF(NO.EQ.1) WRITE(6,1100) NODE(NO),(ICH(I),I=1,18) 00001100
      IF(NO.EQ.2) WRITE(6,1200) NODE(NO),(ICH(I),I=1,18) 00001200
      IF(NO.EQ.3) WRITE(6,1300) NODE(NO),(ICH(I),I=1,18) 00001300
      IF(NO.EQ.4) WRITE(6,1400) NODE(NO),(ICH(I),I=1,18) 00001400
      IF(NO.EQ.5) WRITE(6,1500) NODE(NO),(ICH(I),I=1,18) 00001500
      IF(NO.EQ.6) WRITE(6,1600) NODE(NO),(ICH(I),I=1,18) 00001600
      IF(NO.EQ.7) WRITE(6,1700) NODE(NO),(ICH(I),I=1,18) 00001700
      IF(NO.EQ.8) WRITE(6,1800) NODE(NO),(ICH(I),I=1,18) 00001800
1000 RETURN                                     00001900
1100 FORMAT( 1H / 7X,      A4,IX,':',18A4)      00002000
1200 FORMAT( 8X,'1' / 8X,'1-----',A4,IX,':',18A4) 00002100
1300 FORMAT(15X,'1' / 15X,'1-----',A4,IX,':',18A4) 00002200
1400 FORMAT(22X,'1' / 22X,'1-----',A4,IX,':',18A4) 00002300
1500 FORMAT(29X,'1' / 29X,'1-----',A4,IX,':',18A4) 00002400
1600 FORMAT(36X,'1' / 36X,'1-----',A4,IX,':',18A4) 00002500
1700 FORMAT(43X,'1' / 43X,'1-----',A4,IX,':',18A4) 00002600
1800 FORMAT(50X,'1' / 50X,'1-----',A4,IX,':',16A4) 00002700
      END                                              00002800
           00002900
           00003000
           00003100
           00003200
           00003300
           00003400

```

FILE = J3679.TSSMAC.CNTL

DATE 84/03/16(FRIDAY) TIME 17:00:35 PAGE 0001

```
*****
** COND **
*****
```

```
//JCLG JOB
//***** JOB CONTROL LANGAGE FOR CONDENSE COMMAND *****
//* PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND *
//* BACK-UP FILE NAME                                     *
//* AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND        *
//*                                                       *
//*****                                                       00000010
// EXEC JCLG                                         00000020
//SYSIN DD DATA,DLH='++'
// JUSER ???????,XX.XXXXXX,YYYY.ZZZ                 00000030
T.4 C.1 W.0 I.5 P.0 OPN                           00000040
OPTP PASSWORD=?
// EXEC LMGO,LN='J3679.POOLX',PNH=COND           00000050
//* DATA POOL                                         00000060
//* CHANGE DSN:DATA SET NAME                         00000061
// EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT 00000070
//* BACK-UP FILE                                     00000080
//* CHANGE DSN:DATA SET NAME                         00000090
// EXPAND DISKTN,DDN=FT01F001,DSN='JXXXX.@@BACKUP',UNIT=TSSWK, 00000100
// SPC='500,300'                                     00000110
// EXPAND DISK,DDN=FT02F001                         00000120
//*
//**                                                       00000130
//                                                       00000140
//                                                       00000150
//                                                       00000160
//                                                       00000170
//                                                       00000180
//                                                       00000190
//                                                       00000200
//                                                       00000210
//                                                       00000220
//                                                       00000230
//                                                       00000240
//                                                       00000250
//                                                       00000260
```

```
*****
** HTCOPY **
*****
```

```
//JCLG JOB
//***** JOB CONTROL LANGAGE FOR HTCOPY COMMAND *****
//* PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND *
//* BACK-UP TAPE NAME                                     *
//* AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND        *
//*                                                       *
//*****                                                       00000010
// EXEC JCLG                                         00000020
//SYSIN DD DATA,DLH='++'
// JUSER ???????,XX.XXXXXX,YYYY.ZZZ                 00000030
T.4 C.1 W.0 I.5 P.0 OPN HTU                         00000040
OPTP PASSWORD=?
// EXEC LMGO,LN='J3679.POOLX',PNH=HTCOPY          00000050
//* DATA POOL                                         00000060
//* CHANGE DSN:DATA SET NAME                         00000061
// EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT 00000070
//* BACK-UP TAPE                                     00000080
//* CHANGE DSN:DATA SET NAME                         00000090
//                                                       00000100
//                                                       00000110
//                                                       00000120
//                                                       00000130
//                                                       00000140
//                                                       00000150
//                                                       00000160
//                                                       00000170
//                                                       00000180
//                                                       00000190
```

FILE = J3679.TSSMAC.CNTL

DATE 84/03/16(FRIDAY) TIME 17:00:35 PAGE 0001

FILE = J3679.TSSHAC.CNTL DATE 84/03/16(FRIDAY) TIME 17:00:35 PAGE 0002

```

//*      MTV:VOLUME NUMBER OF A TAPE          00000200
//*      POS:DATA SET POSITION ON A TAPE       00000210
// EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',HTV=??????,HTU=TAPE, 00000220
//      POS=?                                00000230
++                               00000240
//                               00000250

```

\*\*\*\*\*  
\*\* HTSAVE \*\*  
\*\*\*\*\*

```

//JCLG JOB                                     00000010
//*****                                         00000020
//*      JOB CONTROL LANGAGE FOR HTSAVE COMMAND   *
//*                                              * 00000030
//*
//*      PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND   *
//*      BACK-UP TAPE NAME.                                * 00000040
//*      AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND     * 00000050
//*
//*****                                         00000060
// EXEC JCLG                                     00000070
//SYSIN DD DATA,DLH='++'
// JUSER ????????,XX.XXXXXX,YYYY.ZZZ           00000080
T.4 C.1 W.0 I.5 P.0 OPN HTU                   00000090
OPTP PASSWORD=?
// EXEC LHGO,LH='J3679.POOLX',PNH=HTSAVE        00000100
//*      DATA POOL                                 00000110
//*      CHANGE DSN:DATA SET NAME                 00000120
// EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????' 00000130
//*      BACK-UP TAPE                            00000140
//*      CHANGE DSN:DATA SET NAME                 00000150
//*      MTV:VOLUME NUMBER OF A TAPE             00000160
//*      POS:DATA SET POSITION ON A TAPE         00000170
// EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',HTV=??????,HTU=TAPE, 00000180
//      POS=?,DISP='NEW,PASS'                    00000190
// EXPAND DISK,DDN=FT02F001                      00000200
++                               00000210
//                               00000220
//                               00000230
//                               00000240
++                               00000250
//                               00000260

```